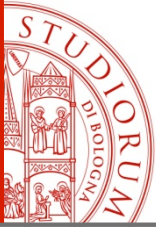


Fondamenti di Ricerca Operativa T-A
a.a. 2014-2014

Simulazione Numerica

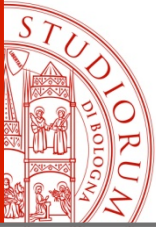
Andrea Lodi, Enrico Malaguti,
Daniele Vigo

Novembre 2014



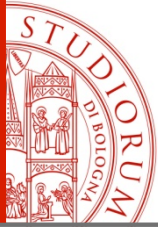
Simulazione numerica

- Tecnica che permette di condurre **esperimenti** su **modelli** (implementati su computer) che “imitano” il comportamento del sistema reale
- Applicata a sistemi molto complessi
 - Dinamici e non stazionari
 - Risorse scarse e sistemi con code
 - Modelli analitici di difficile, se non impossibile, realizzazione
- La simulazione è uno degli strumenti più utilizzati nella pianificazione di sistemi produttivi e logistici



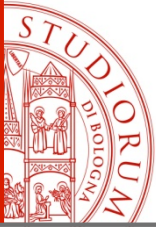
Simulazione numerica (2)

- La simulazione di un sistema richiede una rappresentazione cronologica degli **stati** attraverso cui il sistema evolve lungo il tempo
- Il modello di simulazione è il programma che riproduce il comportamento del sistema:
 - Permettendone una modellazione accurata e dettagliata
 - Permettendo di includere esplicitamente:
 - non-stazionarietà;
 - stocasticità;
 - evoluzione nel tempo.



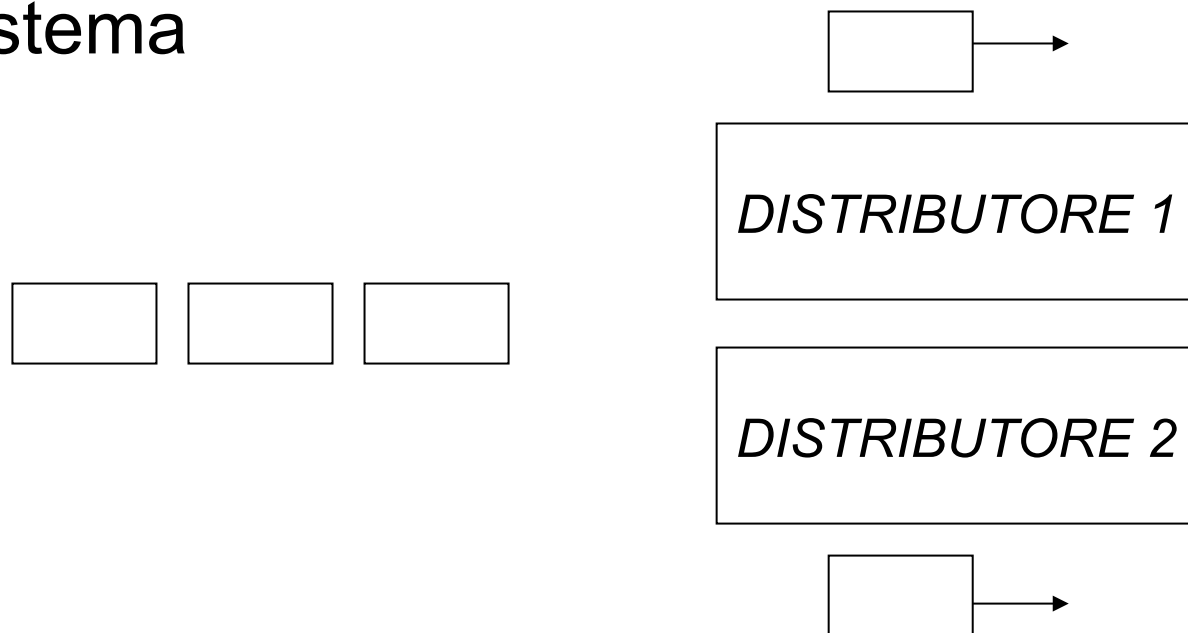
Inconvenienti della simulazione

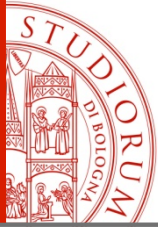
- Non sono modelli analitici: permettono solamente di ottenere stime dei valori che si vogliono osservare;
- I fenomeni stocastici nel modello sono descritti attraverso distribuzioni di probabilità;
- I risultati ottenuti sono “casuali”
 - (simulando più volte lo stesso sistema si ottengono valori differenti)
 - Gli esperimenti devono essere replicati ed i risultati analizzati con metodi statistici



Esempio: stazione di servizio

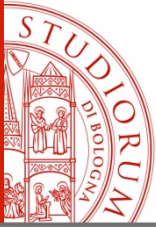
- 2 distributori per il rifornimento (risorse)
- Un'unica coda FIFO per le automobili in attesa (entità)
- Quando il servizio termina le automobili escono dal sistema





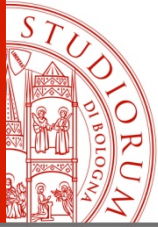
Stazione di servizio (2)

- **Scopo:** valutare il dimensionamento dell'impianto
 - Tempo di attesa delle entità
 - Lunghezza media e massima della coda
 - Grado di saturazione dei distributori
- Qual'è un dimensionamento soddisfacente?
- Identificazione dei “colli di bottiglia” del sistema
- Modifica e verifica delle configurazioni alternative
- Definizione dell'unità di tempo



Stazione di servizio (3)

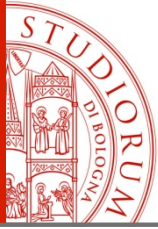
- Se sono già presenti NMAX automobili in coda l'auto in arrivo non si ferma presso la stazione
- Identificare i dati sui tempi caratterizzanti il sistema
 - Frequenza degli arrivi
 - Tempo di servizio
- Osservazione o descrizione tramite variabili casuali caratterizzate dal "tipiche" distribuzioni di probabilità
- Ipotesi:
 - Tempo (t) espresso in minuti
 - Sistema inizialmente ($t = 0$) vuoto e inattivo (libero)



Stazione di servizio (4)

- Esempio di dati di input (noti, osservati) in minuti

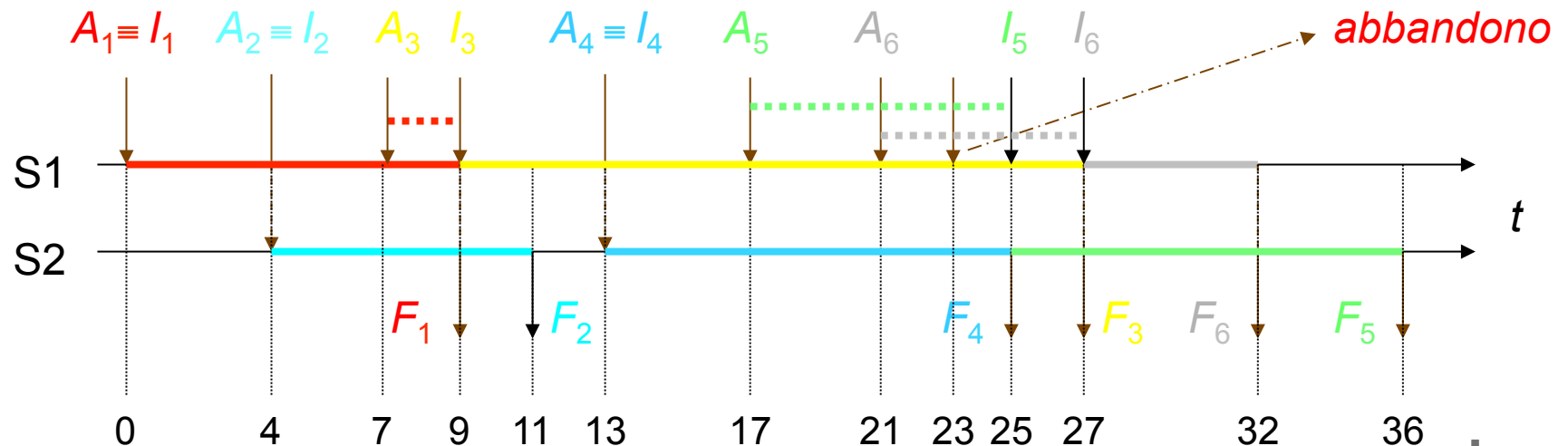
auto	arrivo	interarrivo	t attesa	t servizio	uscita
1	0	—	—	9	9
2	4	4	—	7	11
3	7	3	2	18	27
4	13	6	—	12	25
5	17	4	8	11	36
6	21	4	6	5	32
7	23	2	—	abbandono	23

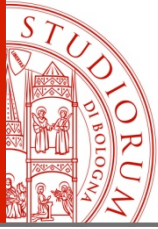


Comportamento dinamico

NMAX = 2

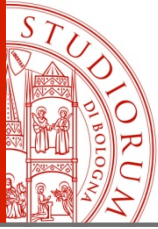
auto	arrivo	interarrivo	t attesa	t servizio	uscita
1	0	—	—	9	9
2	4	4	—	7	11
3	7	3	2	18	27
4	13	6	—	12	25
5	17	4	8	11	36
6	21	4	6	5	32
7	23	2	—	abbandono	23





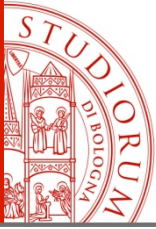
Input del modello

- Dati deterministici:
 - Numero di risorse;
 - Capacità delle code;
 - Tempo di trasferimento, servizio, interarrivo(?).
- Dati stocastici:
 - Tempo di trasferimento, servizio, interarrivo(?);
 - da descrivere attraverso “tipiche” distribuzioni:
 - quali? con quali parametri?
 - “generati” durante la simulazione;
 - input stocastico \Rightarrow output stocastico.



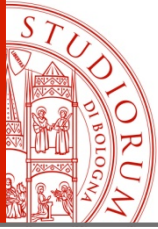
Raccolta dati

- Difficile, costosa, lunga:
 - Il sistema potrebbe non esistere (non osservabile);
 - Dati inesistenti o incompleti (modifica il modello).
- Sensibilità dell'output all'incertezza dell'input
- Correlazione tra modello e dati disponibili
- Costo della raccolta dati da includere nel budget del progetto
- Garbage In, Garbage Out (GIGO)



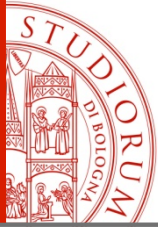
Raccolta dati (2)

- Utilizzo diretto dei dati nella simulazione:
 - Letture da file dei dati osservati (tempi, inter-arrivi,...)
 - Tutti i dati utilizzati sono “veri”
 - Dati non osservati non possono essere generati
 - Può non essere sufficiente per esecuzioni multiple
- Adattare i dati a distribuzioni “tipiche”:
 - Numeri casuali sono generati per i dati
 - L’adattamento dei dati può non essere efficace
 - Dati non osservati possono essere generati
 - Si possono generare dati sufficienti per esecuzioni multiple



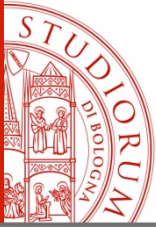
Adattare i dati alle distribuzioni

- Le distribuzioni di probabilità tipiche sono definite analiticamente da formule dipendenti da un numero limitato di parametri
- Dato un insieme di osservazioni
 - Determiniamo la distribuzione (uniforme, esponenziale,...)
 - Determiniamo i parametri
- Valutazione della qualità
 - Scostamento tra valori osservati e generati
 - max likelihood, min squares ...

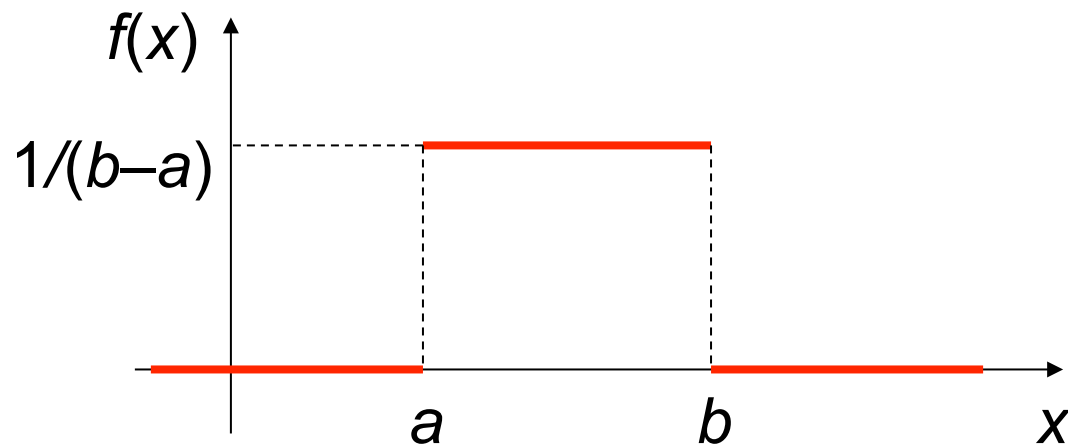


Distribuzioni di probabilità

- Molteplici famiglie di distribuzioni di probabilità hanno rilevanza pratica:
 - discrete r.v.
 - continuous r.v.
- Ogni distribuzione è una famiglia
 - Di distribuzioni di probabilità (discrete r.v.)
 - Di funzioni di densità di probabilità (continuous r.v.)
- Dipendono da un numero limitato di parametri

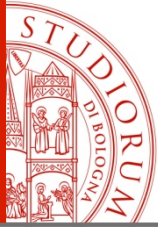


Distribuzione continua uniforme



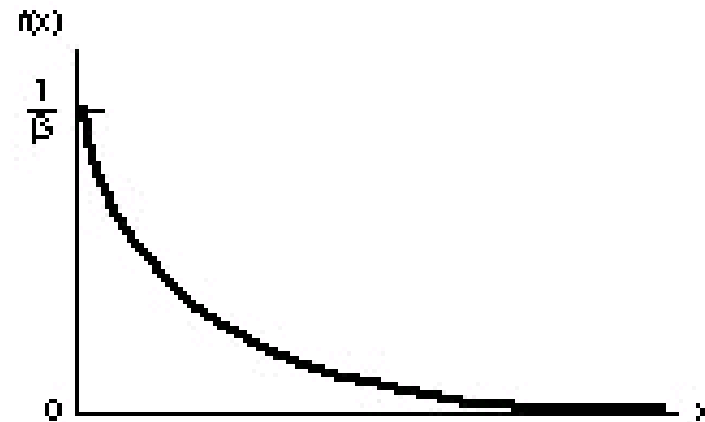
$$f(x) = \begin{cases} 1/(b-a) & x \in [a, b] \\ 0 & \text{altrimenti} \end{cases}$$

- parametri: a, b ; $\mu = (a + b)/2$, $\sigma^2 = (b - a)^2/12$
- Es. peso di un pacchetto di pasta da 1Kg, $a=990\text{g}$, $b=1010\text{g}$

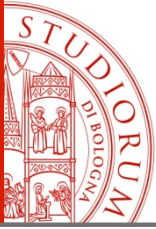


Distribuzione esponenziale

- $f(x) = \beta e^{-\beta x}$ per $x \geq 0$
- parametro: β

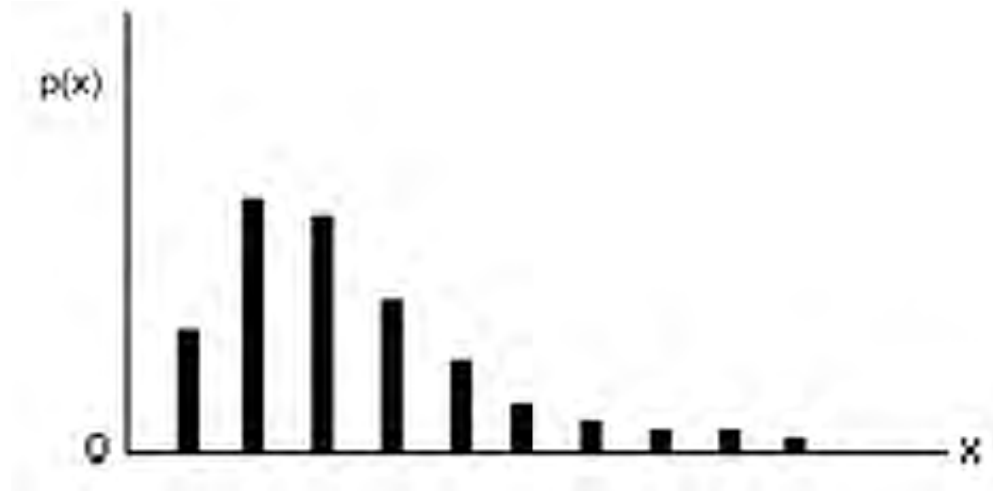


- distribuzione di intervalli temporali in cui avvengono fenomeni “rari” e “indipendenti”
- ex. Arrivo delle automobili alla stazione di servizio
- β = tempo medio di inter-arrivo

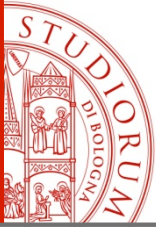


Distribuzione di Poisson (discreta)

- $f(x) = (e^{-\lambda} \lambda^x) / x!$ per $x = 0, 1, 2, 3, \dots, n$; 0 altrimenti
- parametro: λ



- numero di eventi “rari” e “indipendenti” che si verificano nell’intervallo di tempo
- ex. Arrivo delle automobili alla stazione di servizio
- λ = numero medio di arrivi nell’intervallo di tempo

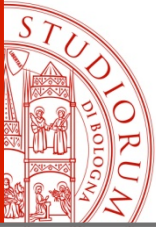


Stazione di servizio: dati

- Arrivo delle automobili:
 - Con distribuzione di Poisson di valor medio λ
- Servizio:
 - Durata distribuita uniformemente in $[T1, T2]$
- Esperimento:

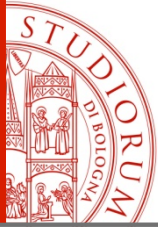
Dati λ , $T1$, $T2$, determina

 - Tempo medio di attesa delle automobili
 - Numero di automobili che abbandonano la stazione di servizio



Modello di simulazione

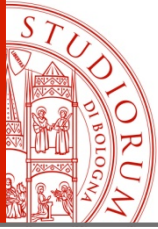
- Suddividi il sistema in un insieme di:
 - Elementi (**entità**) interagenti per l'utilizzo di **risorse** scarse del sistema, cui sono associate **code**
 - Le entità hanno caratteristiche particolari (**attributi**)
- Descrivi il comportamento in termini di:
 - Attività che producono variazioni dello stato del sistema (**eventi**)
 - Percorsi seguiti dalle entità durante la loro vita nel sistema (**processi**)



Elementi del modello (1)

- **Entità**

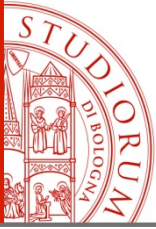
- “attori” che muovono il sistema, cambiano il suo stato e interagiscono con altre entità
- Sono oggetti dinamici: entrano (creazione) e lasciano (distruzione) il sistema
- Normalmente rappresentano entità “reali” (es. automobili, oggetti,...)
- In un dato istante molteplici entità dello stesso tipo “esistono” (sono presenti nel sistema)
- Ci possono essere tipi differenti di entità nello stesso sistema (es. aerei, piloti, bagagli, passeggeri,...)



Elementi del modello (2)

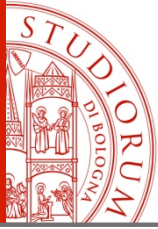
- **Attributi**

- Sono le caratteristiche che descrivono ed identificano le entità
- Tutte le entità dello stesso tipo hanno lo stesso insieme di attributi ma ognuno ha valori differenti
 - Istante di arrivo
 - Istante di consegna
 - Dati per statistiche
- Possono essere considerati come dati “locali” relativi a ogni entità



Elementi del modello (3)

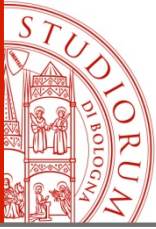
- **Variabili (globali)**
 - Caratteristiche dell'intero modello (non delle entità)
 - Numero di distributori
 - Capacità massima delle code
 - Istante della simulazione
 - Dati globali accessibili da tutte le componenti del modello



Elementi del modello (4)

- **Risorse**

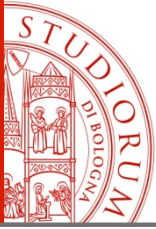
- Quali entità sono in competizione per:
 - distributori
 - spazio, operatori, macchine,...
- Le entità ottengono, utilizzano e rilasciano risorse
- Sono componenti “permanent” del sistema (“vivono” più a lungo delle entità)
- Una risorsa può essere presente in più “copie”
 - Distributori
 - Posti a sedere in un ristorante,...
- Il loro numero può variare durante la simulazione



Elementi del modello (5)

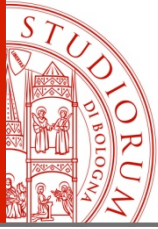
- **Code**

- Sono le “zone” in cui le entità “attendono”, tipicamente la disponibilità delle risorse
- Generalmente sono “connesse” a risorse
- Capacità infinita
- Capacità finita (cosa succede alle entità che arrivano quando una coda è piena?)
- Politica di gestione delle code:
 - FIFO (First In First Out)
 - LIFO (Last In First Out)
 - Ordinata in base ai valori degli attributi delle entità membre



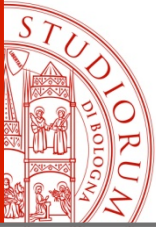
Elementi del modello (6)

- **Statistiche**
 - Memorizziamo i dati che vogliamo osservare
 - Tempo medio nel sistema delle entità
 - Tempo medio in coda delle entità
 - Lunghezza media delle code
 - Numero di entità simulate
 - Ecc.
- Possono essere considerate come variabili globali



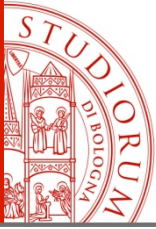
Costruzione del modello (1)

- Richiede un elevato livello di conoscenza del sistema
- Diagramma di flusso per descrivere l'evoluzione del sistema e le relazioni causa-effetto che lo caratterizzano
 - Individuazione dei componenti (**descrizione statica**)
 - Interazione tra i componenti (**descrizione dinamica**)
 - Regole operative che determinano il comportamento del sistema e l'occorrenza degli eventi
- Definizione delle distribuzioni di probabilità



Costruzione del modello (2)

- Un modello molto dettagliato potrebbe non dare più informazioni di un modello semplice
- Se il comportamento di un elemento non può essere descritto in maniera deterministica è meglio utilizzare fenomeni casuali piuttosto che valori medi
- Il modello può essere formalizzato ottenendo un **programma di simulazione** che può essere eseguito su un computer



Costruzione del modello (3)

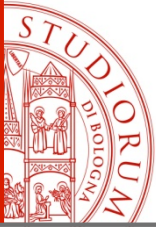
Il programma è attivato utilizzando numeri casuali al fine di generare eventi simulati nel tempo



Ripetendo gli esperimenti possiamo ottenere informazioni statistiche sul comportamento del sistema



Modificando la configurazione possiamo trovare la più soddisfacente

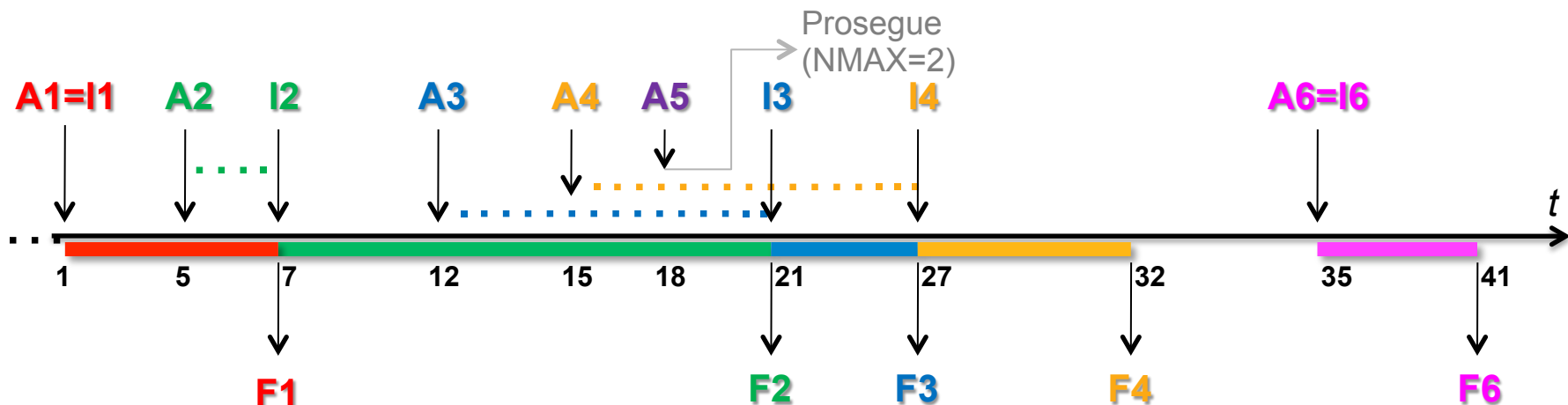


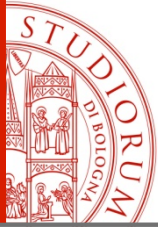
Stazione di Servizio

Descrizione dinamica del sistema

Esempio:

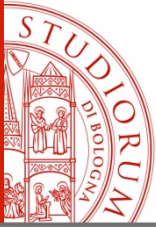
- $NMAX=2$ e
- sistema inizialmente a riposo





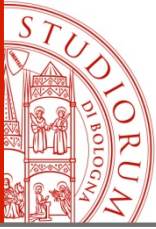
Programmazione degli Eventi

- Evento:
istante in cui avviene
una modifica dello
stato del sistema
(arrivo di un auto, ...)
- Evento:
sottoprogramma
contenente le istruzioni
da eseguire nell'istante
in cui l'evento avviene
- Quando si verifica un evento si considerano
tutte le sue possibili implicazioni sul sistema
(modifica dello stato)



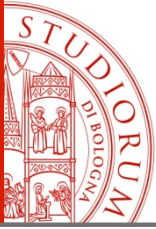
Programmazione degli Eventi (2)

- Ogni evento determina quali eventi debbano avvenire nel futuro e in quale istante (li “innesca”).
- Al termine dell'esecuzione di un evento, il sistema di gestione della simulazione:
 - determina il prossimo evento da eseguire (coda del tempo)
 - aggiorna il contatore del tempo simulato (istante di innesco)
 - manda in esecuzione il prossimo evento



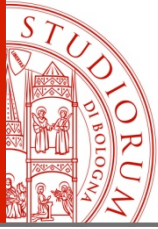
Tempi: Simulato vs Macchina

- Il vantaggio della simulazione al computer è che l'osservazione di un sistema per ore o giorni avviene in secondi o minuti
- Questo è possibile perché esiste una completa distinzione tra
 - tempo “simulato”, e
 - tempo “macchina”



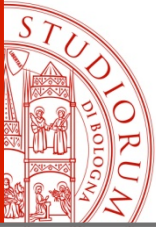
Tempi: Simulato vs Macchina (2)

- Gli eventi (sottoprogrammi da eseguire) sono inseriti nella cosiddetta **CODA DEL TEMPO**
- Ad ognuno di essi, ad es. E, è associato un orario di esecuzione (come tempo simulato), s(E)
- Alla fine dell'esecuzione dell'evento E, il computer ha completa conoscenza degli eventi innescati e “pesca” dalla CODA DEL TEMPO il prossimo da eseguire, ad es. P.



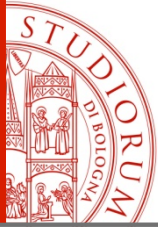
Tempi: Simulato vs Macchina (3)

- Quindi per l'orologio della simulazione, l'orario viene spostato da $s(E)$ a $s(P)$ artificialmente e in maniera istantanea (discreta)
- Il tempo macchina avanza invece in maniera continua relativamente al tempo effettivo di esecuzione delle istruzioni nei sottoprogrammi eventi sul computer



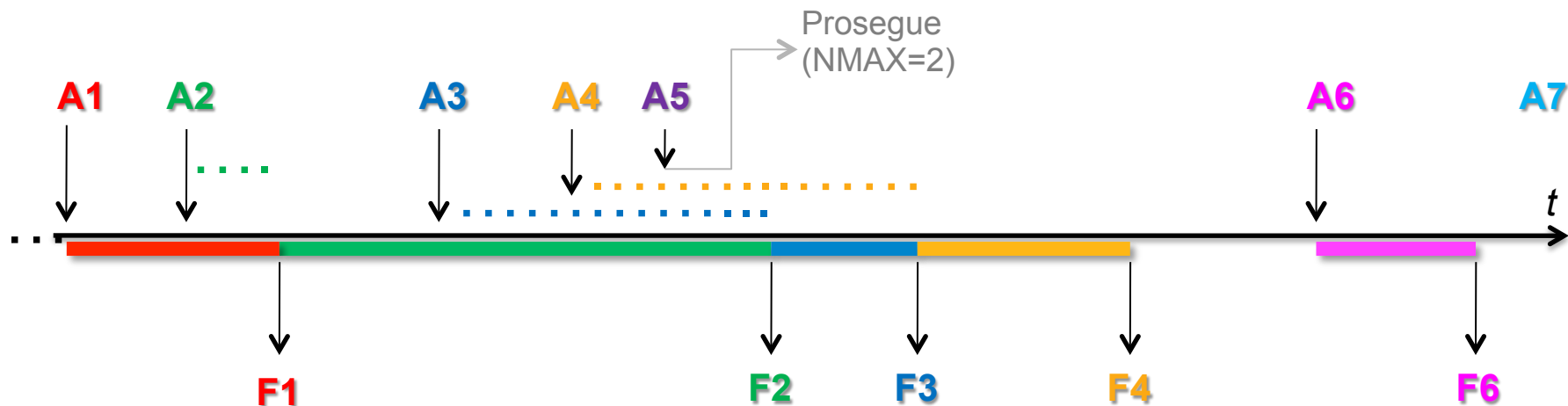
Tempi: Simulato vs Macchina (4)

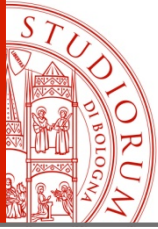
- Il tempo simulato totale è pari all'istante di esecuzione dell'ultimo evento e non dipende dal computer che viene usato per la simulazione
- Il tempo macchina totale è invece la somma dei tempi di esecuzione dei sottoprogrammi e delle procedure e quindi dipende dal computer



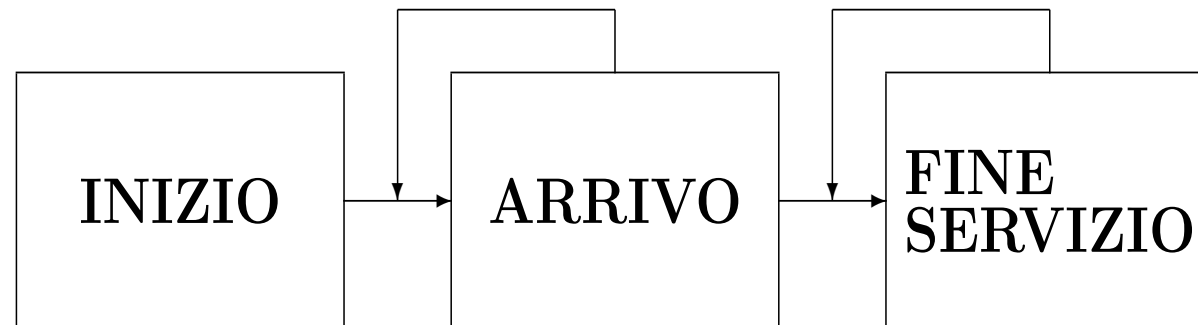
Stazione di Servizio

Logica degli Inneschi

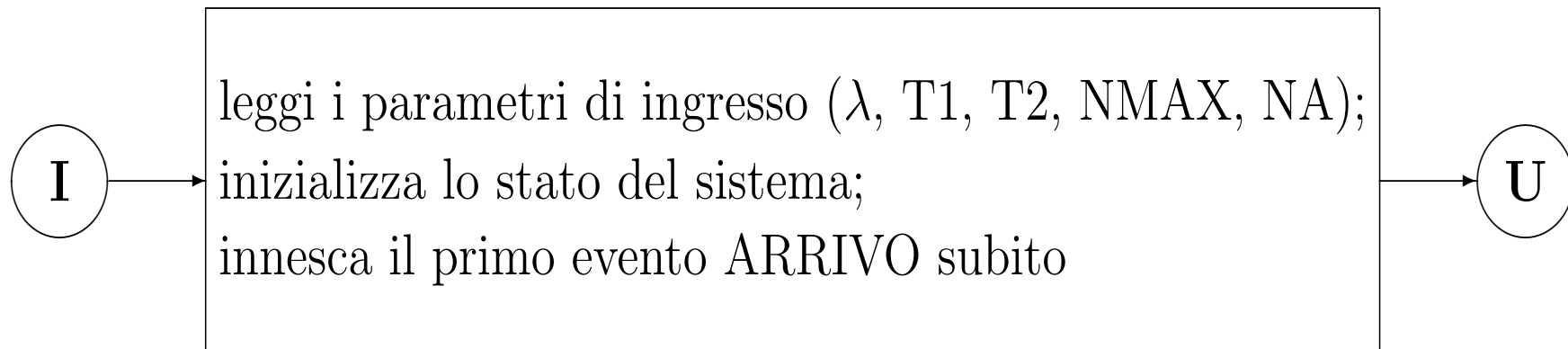


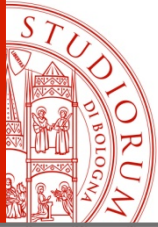


Stazione di Servizio (1)

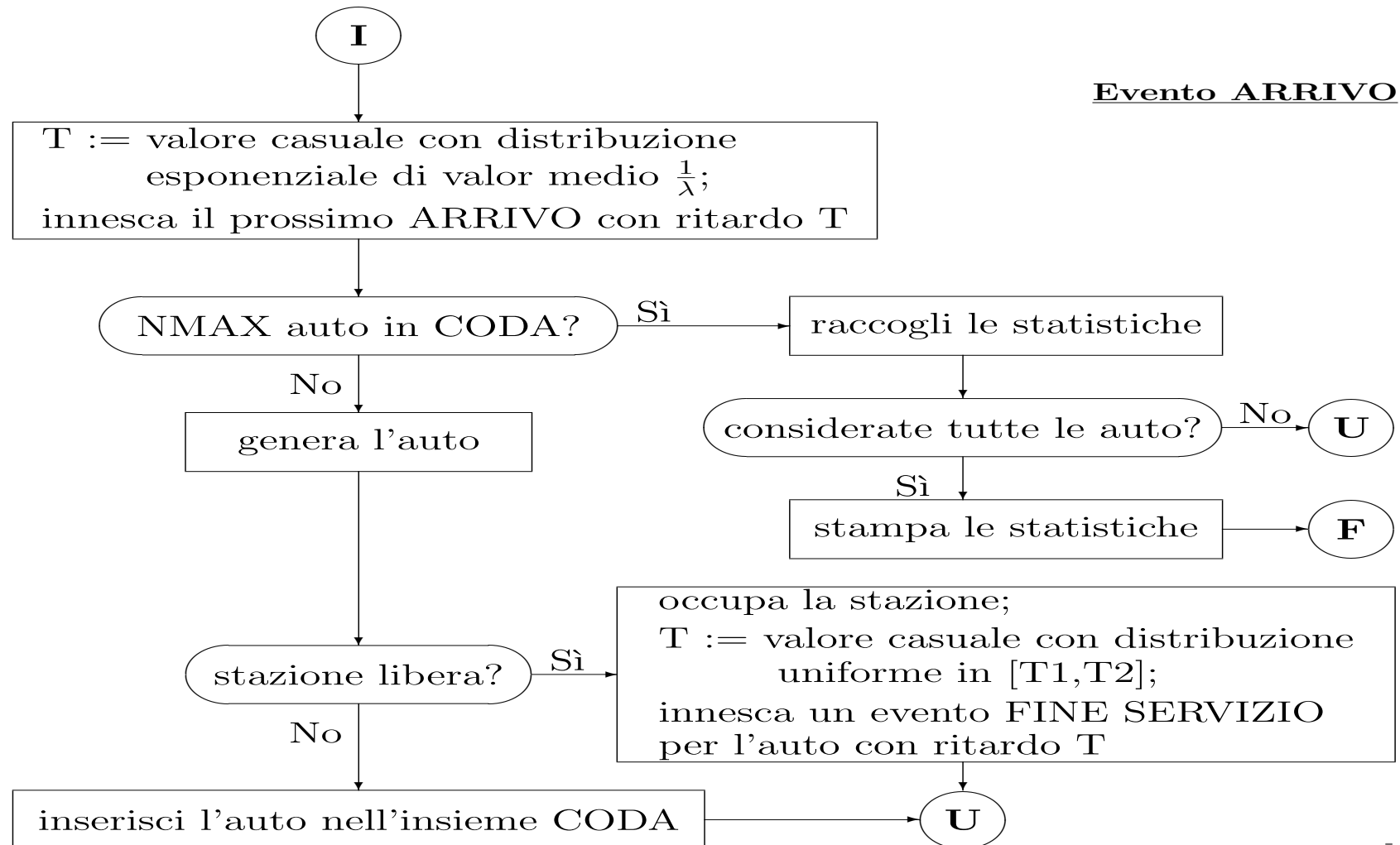


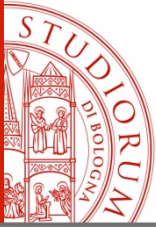
Evento INIZIO





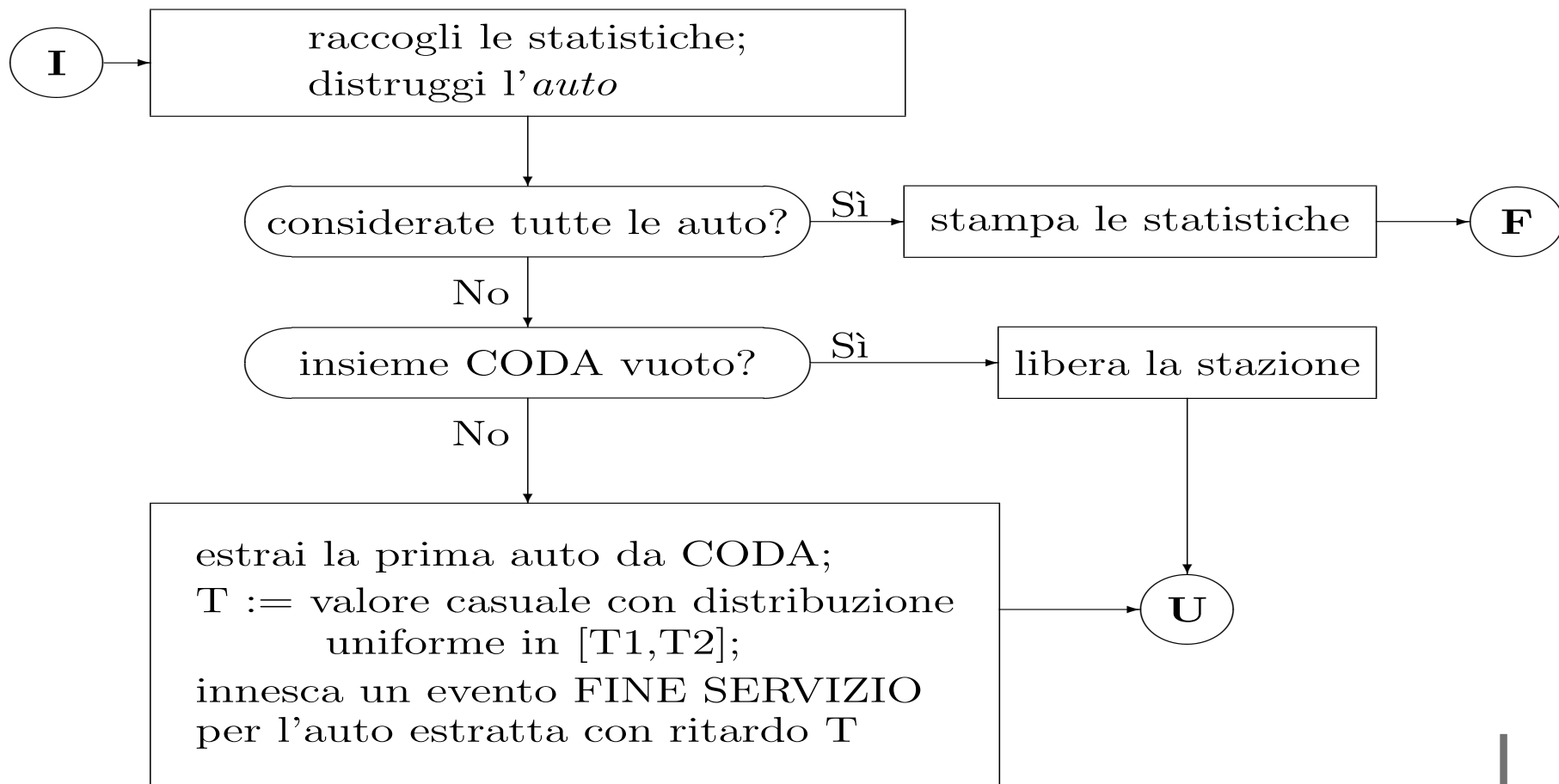
Stazione di Servizio (2)

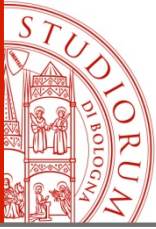




Stazione di Servizio (3)

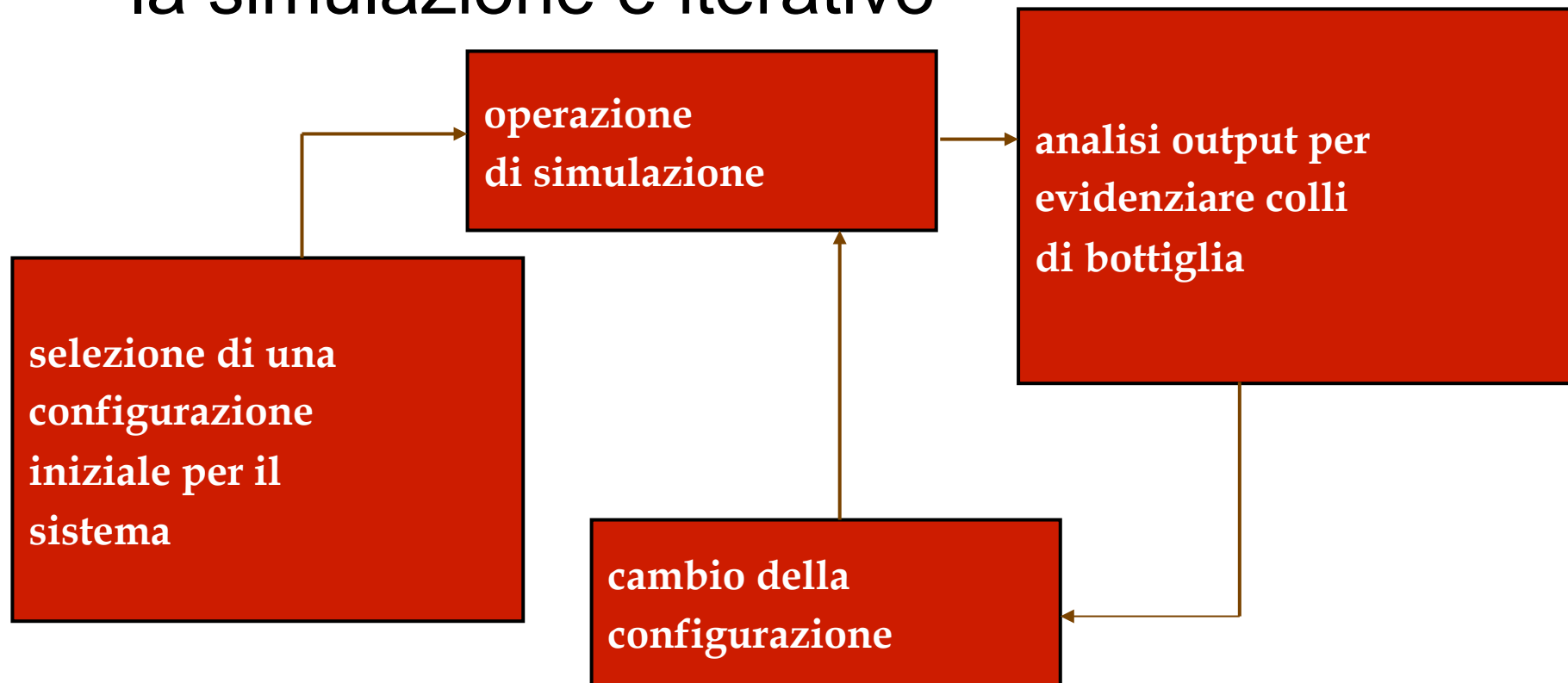
Evento FINE SERVIZIO
per l'*auto*

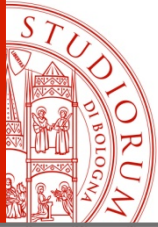




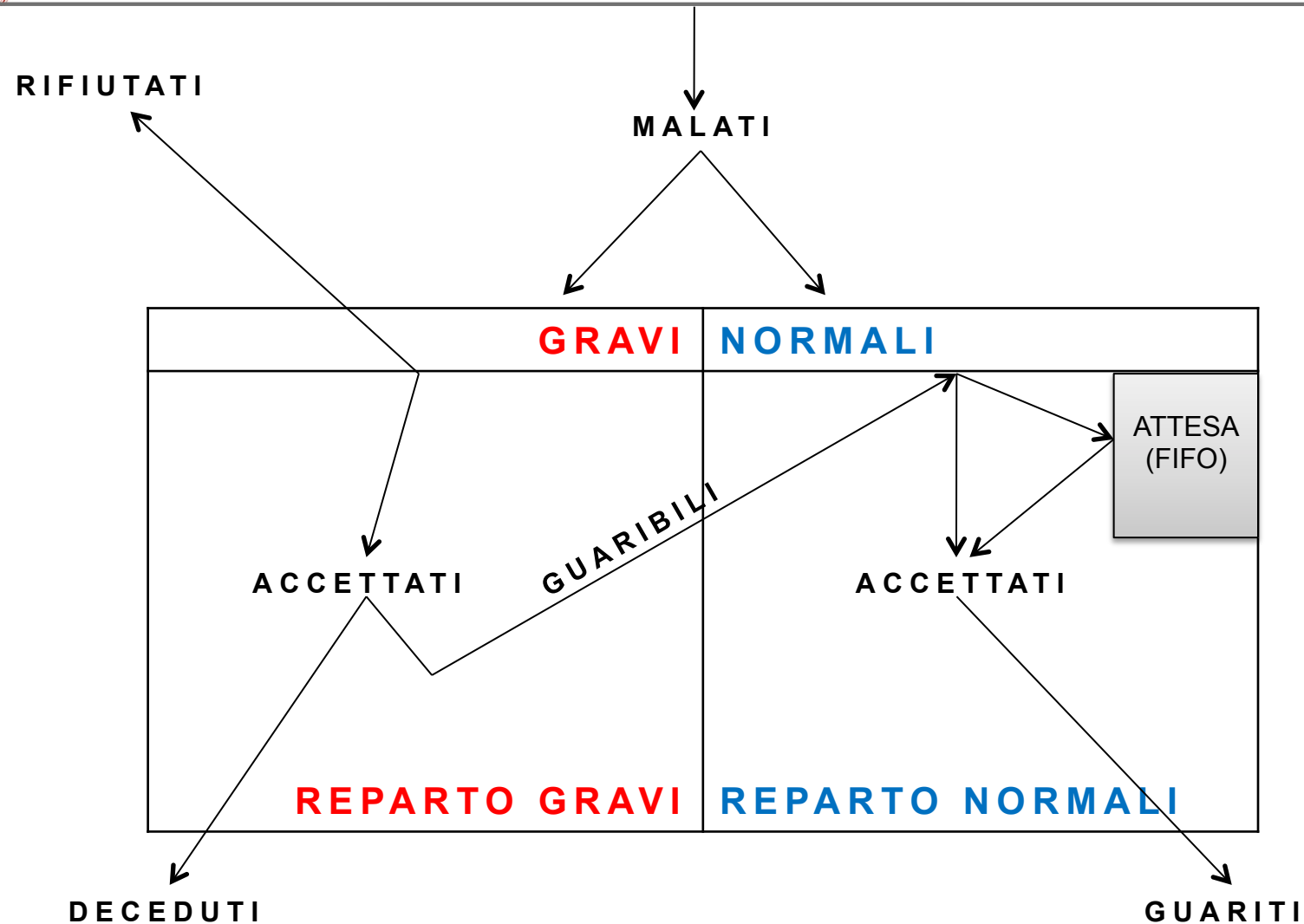
Esperimenti sul modello

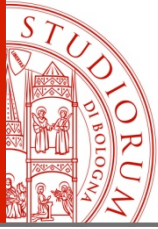
- Il processo di ottimizzazione attraverso la simulazione è iterativo





Sistema ospedaliero





Ospedale, Descrizione (1)

Numero letti: **reparto Gravi**: NLG, **reparto Normali**: NLN.

Arrivo malati: distribuzione di Poisson con valor medio λ .

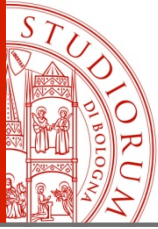
Gravi con probabilità PG, **normali con probabilità 1-PG**.

Grave rifiutato se non vi sono letti liberi nel reparto Gravi.

Durata cura gravi: distribuzione uniforme in [DMIG,DMAG].

Esito cura gravi: positivo con probabilità PS, negativo con probabilità 1-PS; se positivo, passa al reparto Normali.

Durata cura normali: distribuzione uniforme in [DMIN,DMAN].

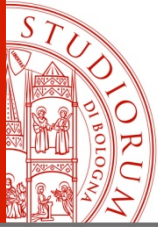


Ospedale, Descrizione (2)

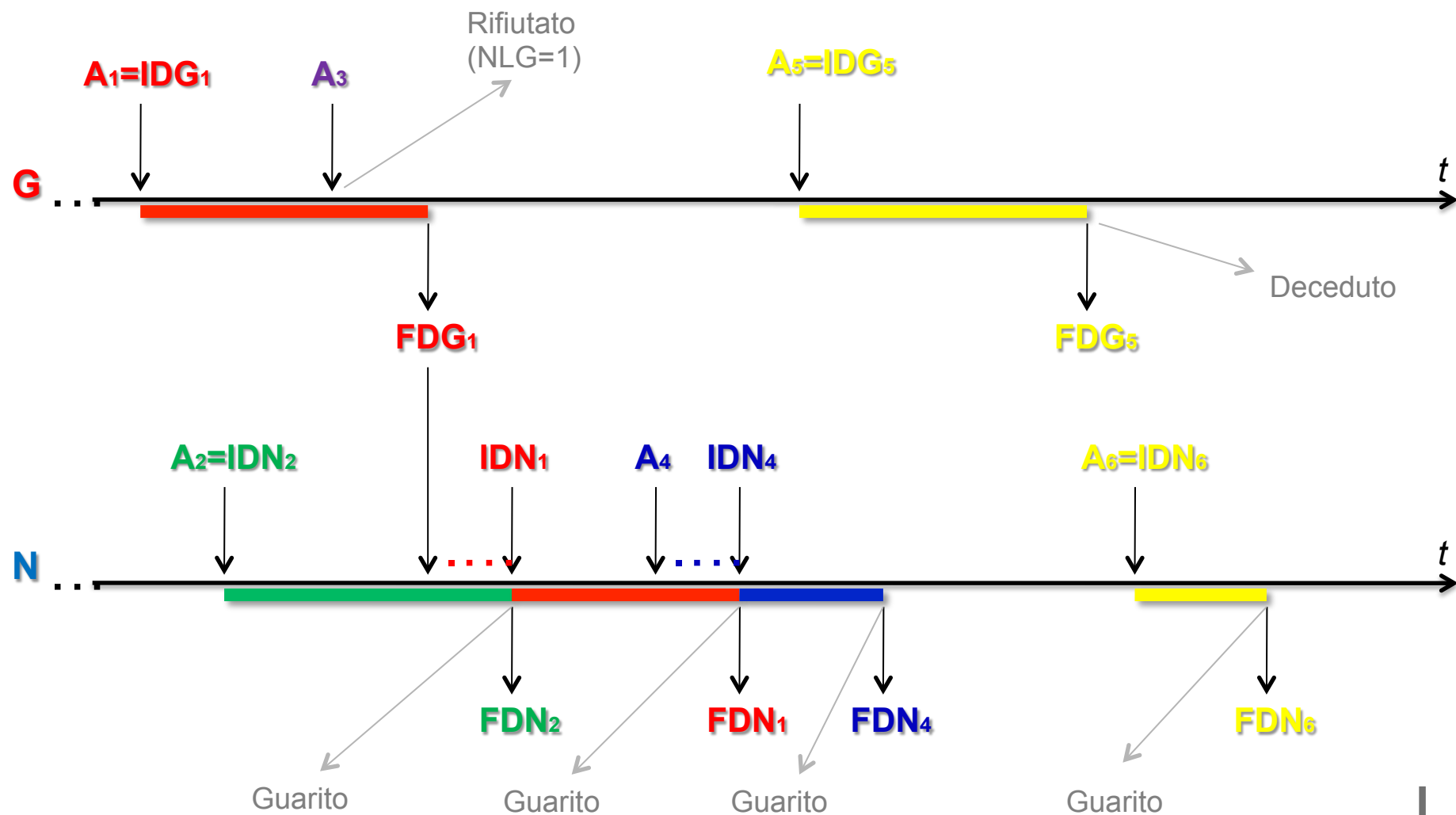
La simulazione termina quando è stato esaminato un numero prefissato NT di malati.

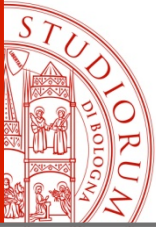
Relativamente agli NT malati usciti dal sistema, determinare la quantità:

- Tempo medio trascorso in coda dai malati che hanno atteso in coda.
- Tempo medio trascorso nel sistema dai malati gravi (deceduti o guariti).
- Tempo medio trascorso nel sistema dai malati normali.
- Percentuale di malati gravi rifiutati.

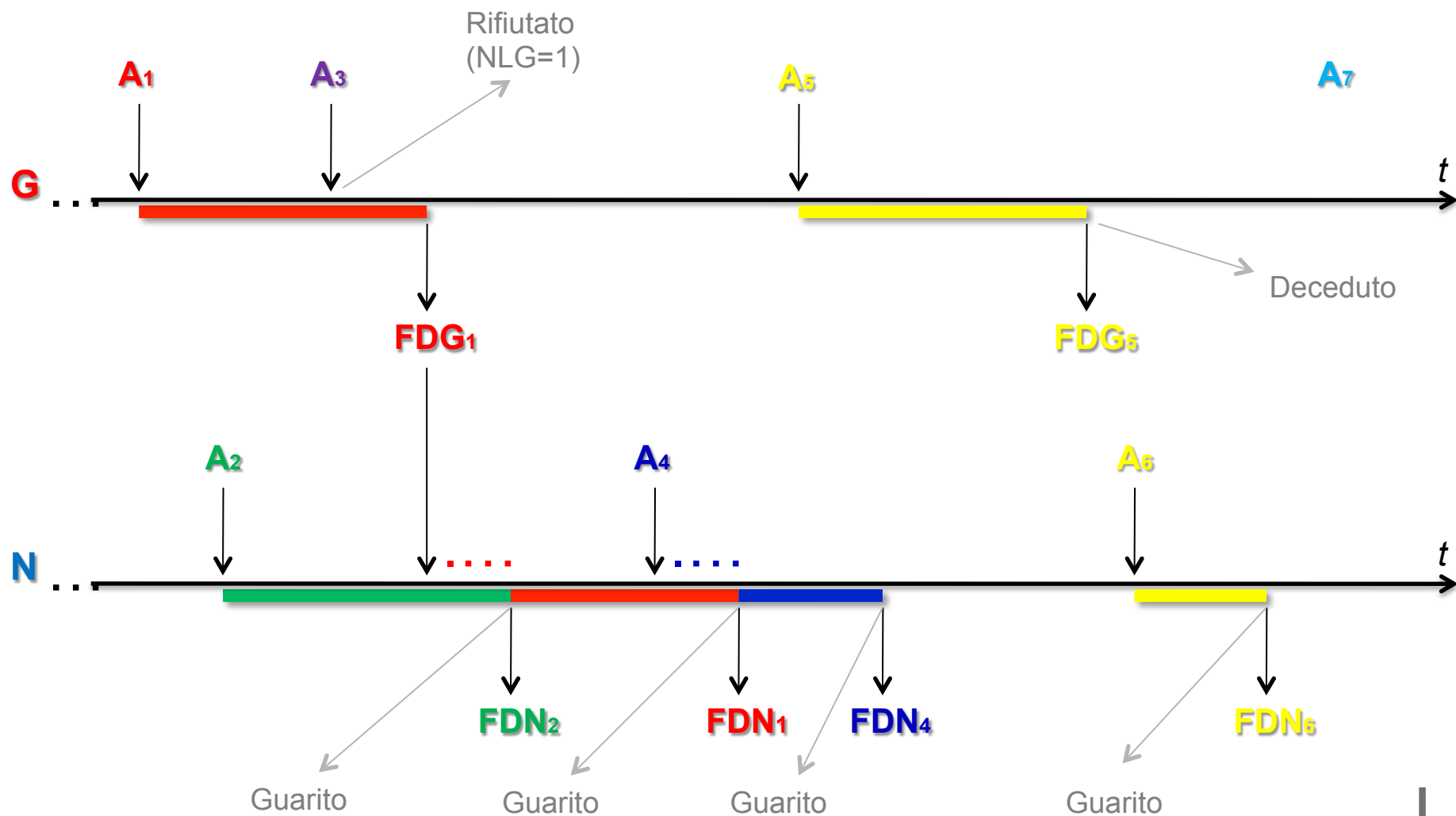


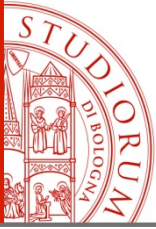
Ipotesi: $NLG = NLN = 1$, sistema inizialmente a riposo:





LOGICA DEGLI INNESCHI

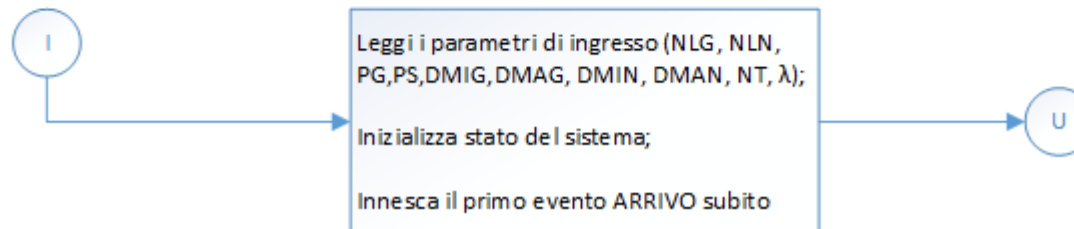


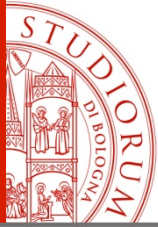


Ospedale (1)



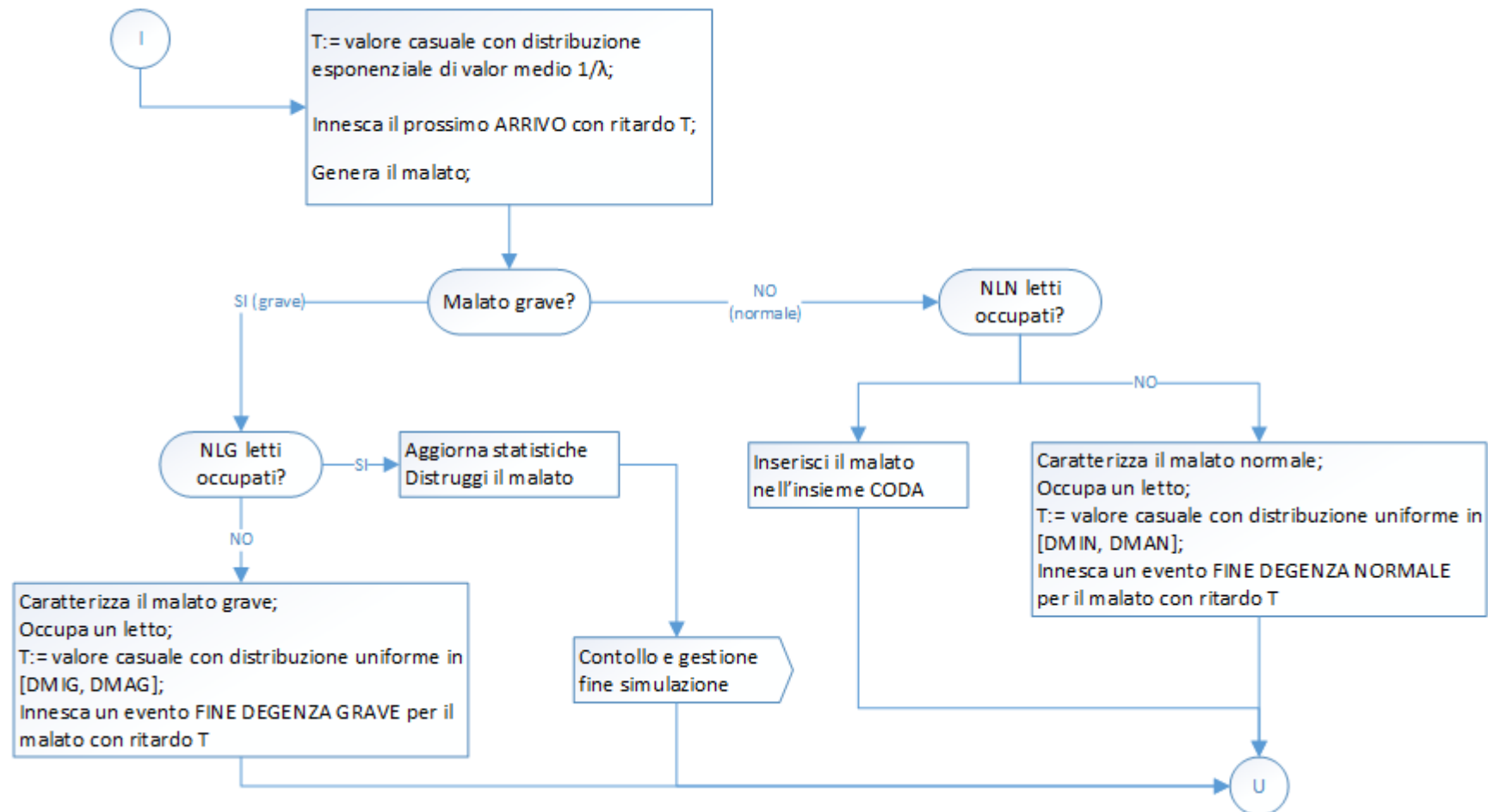
Evento INIZIO

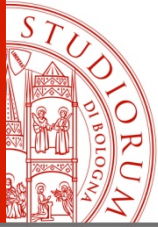




Ospedale (2)

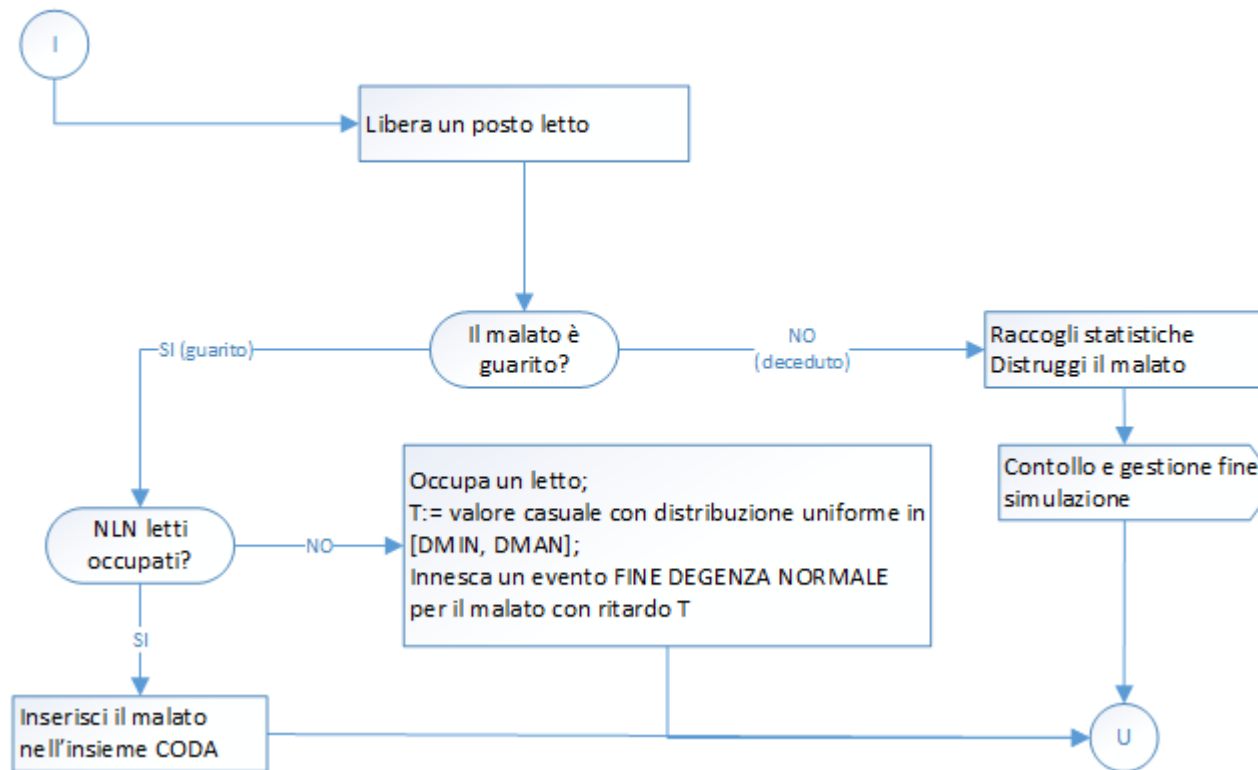
Evento ARRIVO

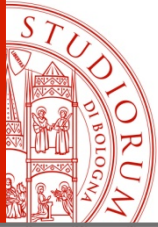




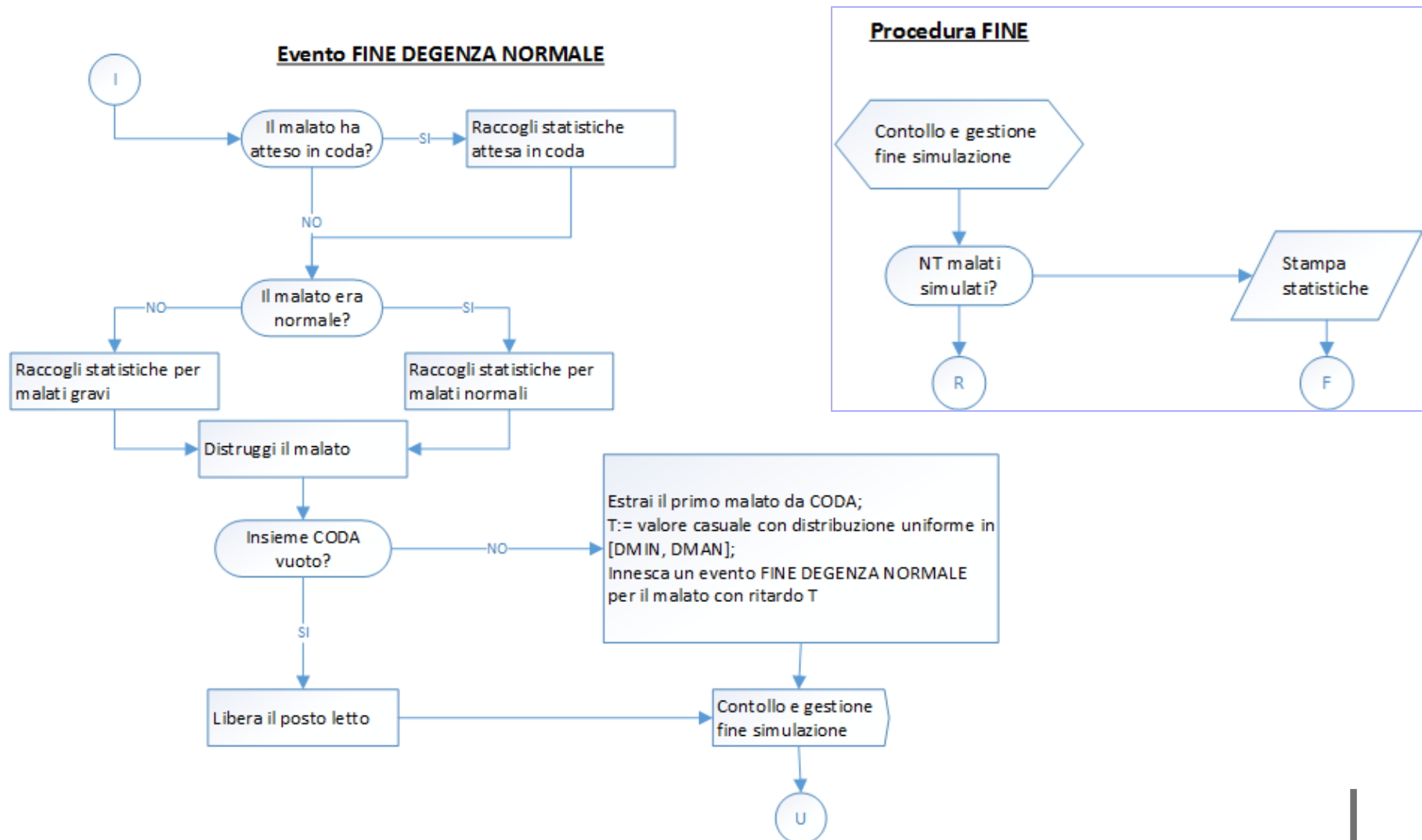
Ospedale (3)

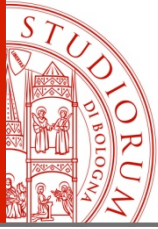
Evento FINE DEGENZA GRAVE





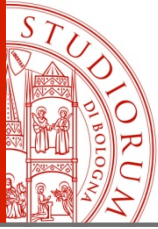
Ospedale (4)





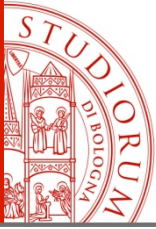
Trasmissione delle Informazioni

- È facile vedere che le versioni descritte degli esempi della Stazione di Servizio e dell'Ospedale non sono abbastanza dettagliate da poter essere “implementate”
- È quindi necessario definire l'aspetto di come le informazioni vengono trasmesse tra il sistema e gli eventi e tra gli eventi stessi



Event Notice e Puntatori

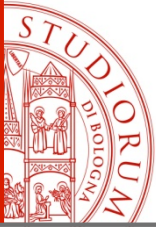
- **Event Notice:**
 - richiesta di esecuzione di un evento
 - include eventuali parametri (attributi)
- Le Event Notice come le Entità sono rappresentate mediante aree di memoria che ne contengono le loro caratteristiche e che sono identificate dai “**puntatori/riferimenti**” = indirizzo dell’area di memoria



Sintassi: Descrizione statica

Entità temporanee: es. AUTO

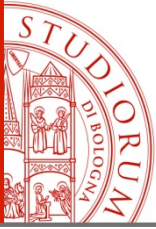
- “crea AUTO”:
 - riserva un’area di memoria per una nuova entità della classe AUTO e ne memorizza il puntatore nella variabile (locale) AUTO
- “distruggi AUTO”:
 - rilascia l’area di memoria puntata da AUTO
- Attributi (descrittivi, dati per statistiche, ...)
 - es. TARGA(AUTO), TIC(AUTO), TC(AUTO), ...



Sintassi: Entità permanenti

Entità permanenti: es. il “Sistema”

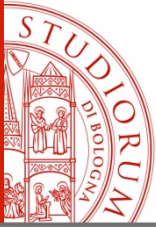
- Attributi:
 - variabili globali (stato, contatori, ...)
- Parti del sistema (con coda) presenti in più esemplari (Erogatori, Macchine, ...)
 - attributi e code con indice (STATO(I), CODA(I))
- “crea K entità Erogatore” (nell’evento INIZIO):
 - riserva K aree di memoria per gli attributi e
 - definisce le K code



Sintassi: Code

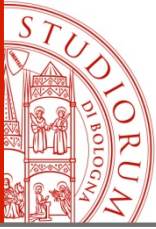
Insiemi (Code):

- quali entità contengono e regola di uscita (FIFO, LIFO, “ordinate”)
 - “inserisci AUTO in CODA”,
 - “estrai la prima AUTO da CODA”,
 - “estrai AUTO da CODA”.
- le code memorizzano i puntatori/riferimenti alle entità che contengono



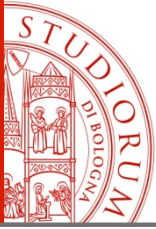
Sintassi: Descrizione dinamica

- Evento:
 - sottoprogramma che viene mandato in esecuzione in base all'evoluzione dinamica del sistema
- Variabile globale TIME.V:
 - istante corrente del tempo simulato
- Richiesta di esecuzione di un evento:
 - event notice
 - area di memoria che ne contiene gli attributi



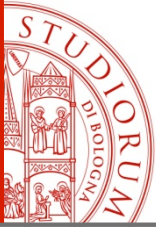
Sintassi: Eventi (1)

- “crea FSER”:
 - riserva un’area di memoria per un’event notice della classe FSER e ne memorizza il puntatore nella variabile FSER
- “innesca FSER con ritardo T”, oppure
- “innesca FSER a $\text{TIME.V} + T$ ”:
 - inserisce nella coda del tempo il puntatore FSER all’event notice di classe FSER, il cui evento associato sarà eseguito tra T unità di tempo



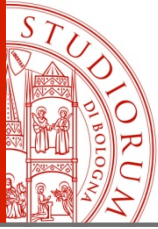
Sintassi: Eventi (2)

- “cancella FSER”:
 - rimuove dalla coda del tempo l’event notice il cui puntatore è in FSER
- “distruggi FSER”:
 - rilascia l’area di memoria dell’event notice il cui puntatore è in FSER



Sintassi: Eventi (3)

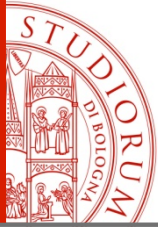
- Gli eventi possono avere attributi:
 - es. puntatore all'AUTO che termina il servizio, AUTO(FSER)
- Eventi interni:
 - innescati da altri eventi (es. ARRIVO, FSER, ...)
- Eventi esterni:
 - innescati dall'esterno (es. INIZIO)
- Procedure:
 - per isolare parti di codice comuni a più eventi



Esecuzione della simulazione (1)

1. Inizializzazione sistema

- Coda del tempo (vuota):
 - struttura che memorizza i puntatori alle event notice degli eventi da eseguire in ordine crescente d'istante di innesco
- Pone $\text{TIME.V} := 0$
 - istante attuale di tempo simulato (variabile globale)
- Creazione delle entità permanenti:
 - code (vuote), attributi



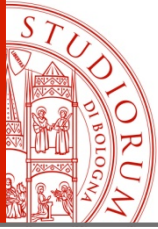
Esecuzione della simulazione (2)

2. Innesco eventi esterni

- Evento INIZIO:
 - inizializza variabili di stato e contatori
 - crea e innesca uno o più eventi interni (es. ARRIVO)

3. Esecuzione eventi interni

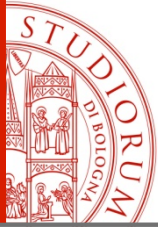
- all'istante di innesco



Esecuzione della simulazione (3)

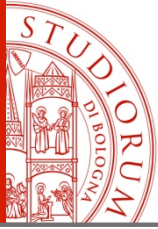
- **while** (coda del tempo non vuota) **do**
 - 3.1 estrae dalla coda il puntatore alla prossima event notice
 - 3.2 pone $\text{TIME.V} := \text{istante di innesco}$
 - 3.3 definisce una variabile locale con lo stesso nome dell'evento e vi memorizza il puntatore all'event notice corrente
 - 3.4 Esegue le istruzioni associate all'evento
 - “Fine” F : termina la simulazione
 - “Uscita” U : prosegue

4. Stampa dei risultati delle osservazioni statistiche



Stazione di Servizio

- Siamo a questo punto pronti a procedere allo sviluppo della versione completa dell'esempio della Stazione di Servizio
- Questa versione (finale) include una trasmissione delle informazioni completa e corretto



Trasmissione di informazioni

1) Attributi permanenti (variabili globali):

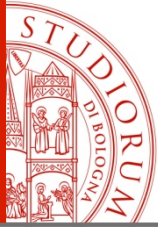
- TTC azzerato in INIZIO,
 aggiornato in FSER,
 utilizzato in FINE.

2) Attributi di event notice:

- AUT(FSER) definito in ARRIVO e FSER
 utilizzato in FSER.

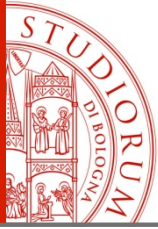
3) Attributi di entità inserite e rimosse da insiemi:

- es. AUTO inserita in CODA in ARRIVO,
 rimossa da CODA in FSER,
 dove si può utilizzare TIC(AUTO)



Trasmissione di informazioni

- Non esistono altri meccanismi di trasmissione di informazioni tra eventi
- Ipotesi:
 - tutte le variabili usate negli eventi sono locali: le relative aree di memoria sono rilasciate all'uscita
 - anche le variabili che si chiamano come le classi di entità ed eventi (AUTO, FSER, ARRIVO ...)



Stazione di Servizio: Descrizione statica (1)

Determinare, relativamente ad NA auto uscite dal sistema:

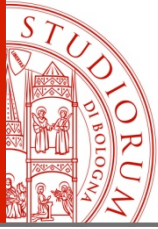
- 1) percentuale di auto che non si fermano
- 2) tempo medio trascorso in coda dalle auto servite

- **Entità temporanee: Attributi:**

AUTO

TIC (istante di ingresso in coda);

TC (tempo trascorso in coda);



Stazione di Servizio: Descrizione statica (2)

- Entità permanenti: Attributi:

Sistema

Dati di ingresso

λ , T1, T2, NMAX, NA;

Raccolta statistiche

NAS = numero auto simulate,

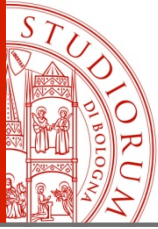
NAR = numero auto che non si fermano,

TTC = tempo totale in coda,

Stato del sistema

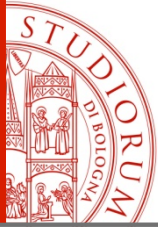
NAC = numero auto in coda,

$$\text{STATO} = \begin{cases} 0 & \text{(stazione libera)} \\ 1 & \text{(stazione impegnata)} \end{cases}$$

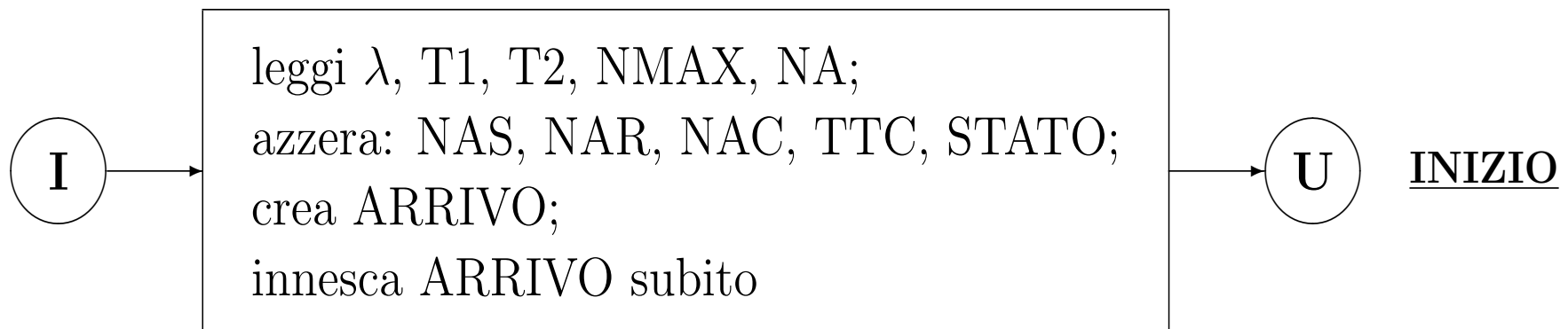
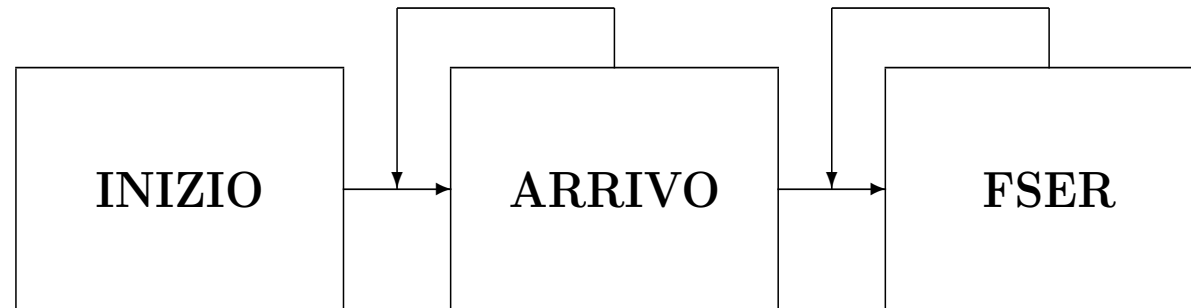


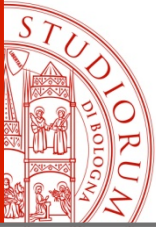
Stazione di Servizio: Descrizione statica (3)

- **Insiemi:** CODA
Ordinamento: FIFO
Entità membre: AUTO
Entità proprietaria: Sistema
- **Eventi:** INIZIO
ARRIVO
FSER
Tipo: esterno
interno
interno
Attributi: —
—
AUT

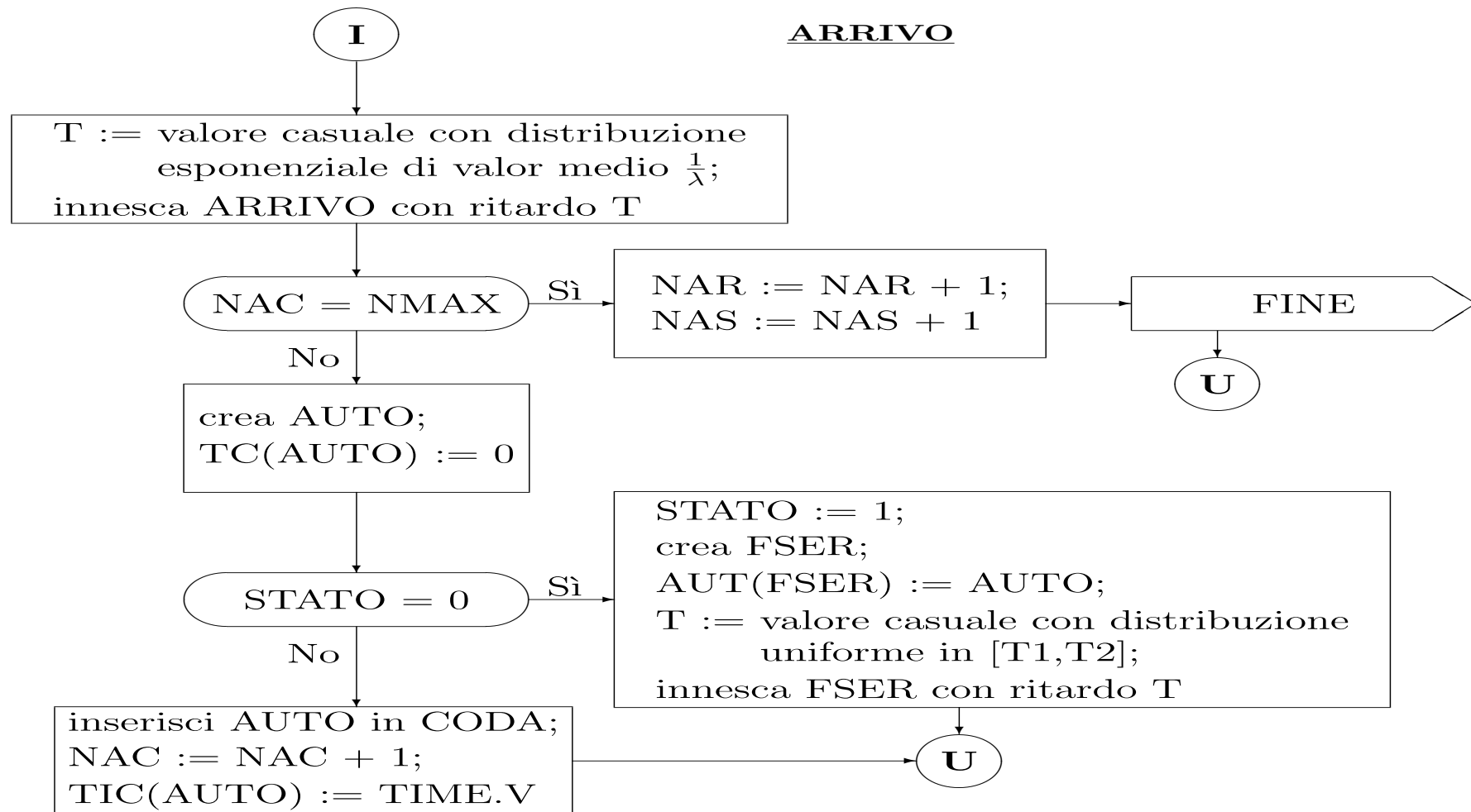


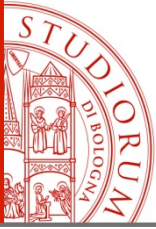
Stazione di Servizio, diagramma di flusso (1)



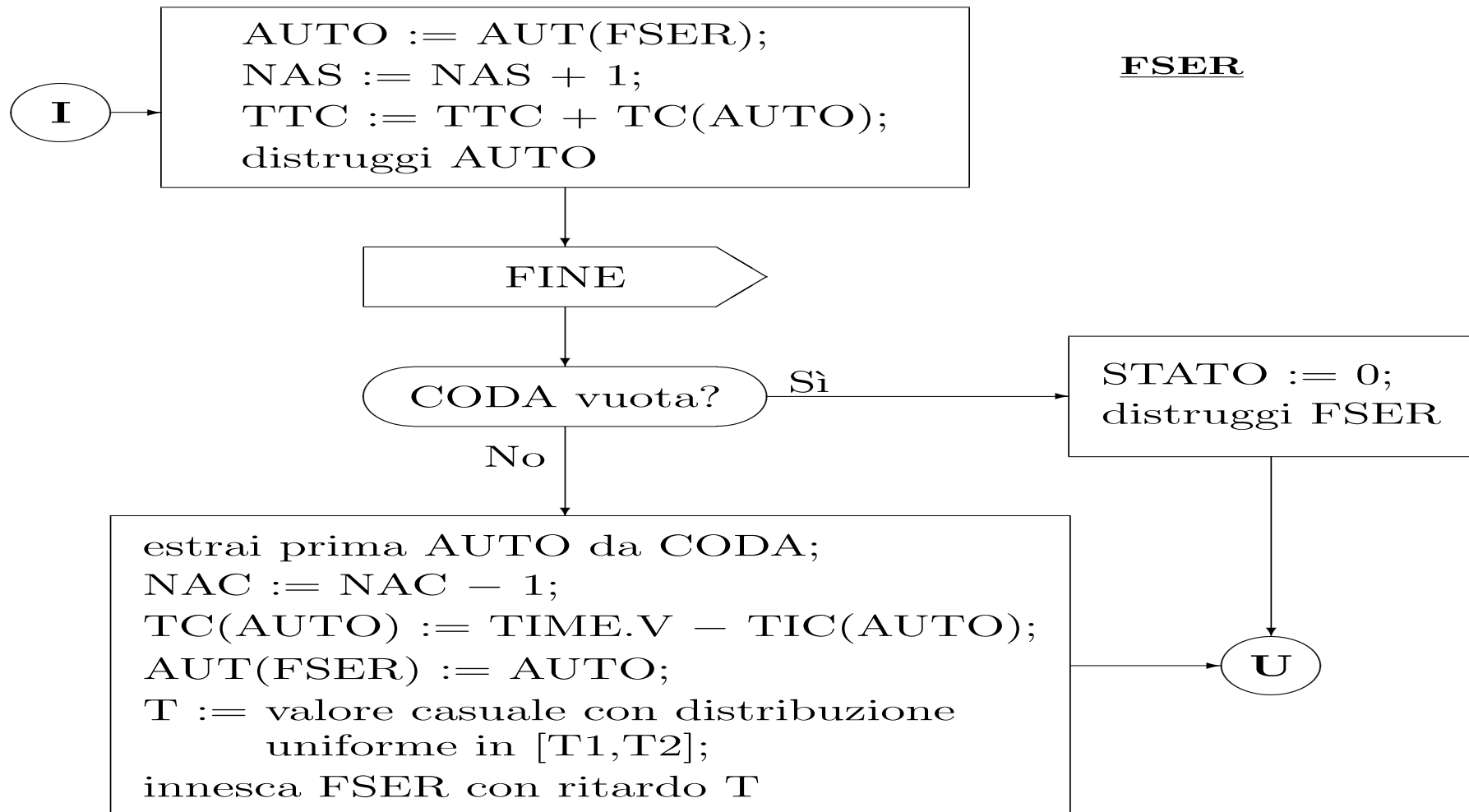


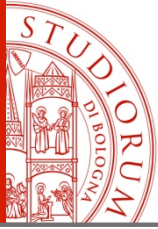
Stazione di Servizio, diagramma di flusso (2)



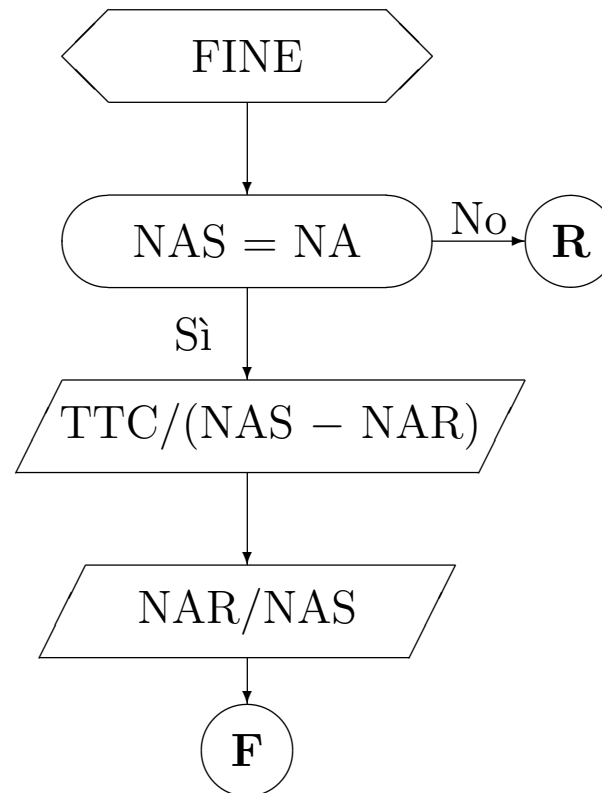


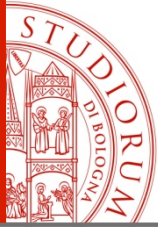
Stazione di Servizio, diagramma di flusso (3)





Stazione di Servizio, diagramma di flusso (4)



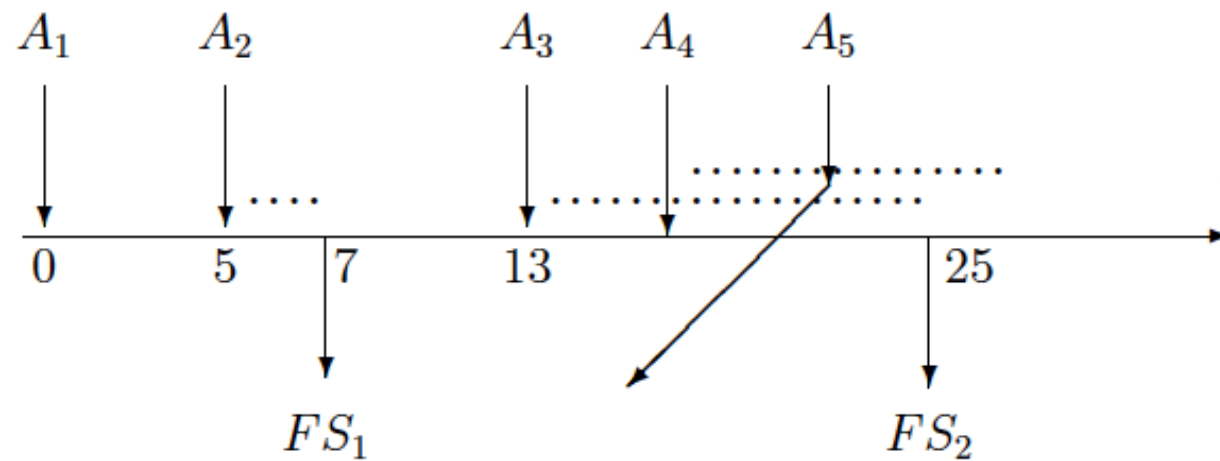


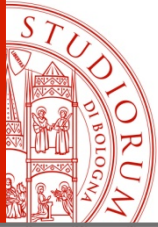
Informazioni e aree di memoria (1)

Stazione di servizio;

ogni AUTO ha due attributi: TIC, TC (istante di ingresso e tempo totale in coda)

FSER ha l'attributo AUT (puntatore all'auto)





Informazioni e aree di memoria (2)

Alla fine di A_1 ($t = 0.0$)	Dopo A_1 (inizio di A_2)	Alla fine di A_2 ($t = 5.0$)	All'inizio di FS_1 ($t = 7.0$)
AUTO → 550	AUTO → ?	AUTO → 555	AUTO → ?
FSER → 800	FSER → ?	FSER → ?	FSER → 800
550 (TIC) ?	550 ?	555 5.0	550 ?
551 (TC) 0.0	551 0.0	556 0.0	551 0.0
800 (AUT) 550	800 550	CODA 555	800 550

nella coda degli
eventi innescati:

ARRIVO	ARRIVO	FSER	ARRIVO	FSER
700	700	800	700	800
0.0	5.0	7.0	13.0	25.0
A_1	A_2	FS_1	A_3	FS_2

dopo $AUTO := AUT(FSER)$

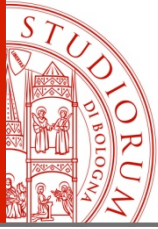
AUTO → 550
...

dopo "estrai prima AUTO" (punt. 555)

AUTO → 555
...

dopo $AUT(FSER) := AUTO$

800 → 555



Ospedale: Descrizione statica (1)

- Entità temporanee: Attributi:

MALATO

TIS (istante di ingresso nel sistema);

TIC (istante di ingresso in coda, se è in coda
tempo trascorso in coda, se non è in coda);

$$\text{TIPO} = \begin{cases} 1 & \text{(normale)} \\ 2 & \text{(grave)} \end{cases}$$

- Entità permanenti: Attributi:

Sistema

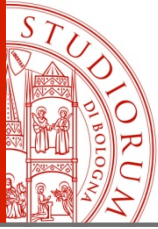
Numero di letti occupati nel Reparto Gravi

Numero di letti occupati nel Reparto Normali

Variabili per raccolta statistiche

...

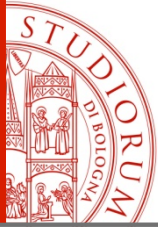
Dati di ingresso



Ospedale: Descrizione statica (2)

- **Insiemi:** CODA **Ordinamento:** FIFO **Entità membre:** MALATO **Entità proprietaria:** Sistema
- **Eventi:**

	Tipo:	Attributi:
INIZIO	esterno	—
ARRIVO	interno	—
FDG	interno	MAL
FDN	interno	MAL



Ospedale: Attributi del sistema (1)

- *Dati di ingresso:*

λ = numero medio di arrivi nell'unità di tempo

NLG = numero letti nel reparto malati gravi

NLN = numero letti nel reparto malati normali

PG = probabilità che un malato sia grave

PS = probabilità che un malato grave guarisca

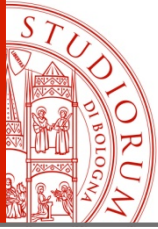
DMIG = durata minima di una degenza grave

DMAG = durata massima di una degenza grave

DMIN = durata minima di una degenza normale

DMAN = durata massima di una degenza normale

NT = numero di malati da simulare



Ospedale: Attributi del sistema (2)

- *Stato del sistema:*

NLOG = numero letti occupati nel reparto malati gravi

NLON = numero letti occupati nel reparto malati gravi

- *Raccolta statistiche:*

NMR = numero di malati gravi rifiutati

NMD = numero di malati gravi deceduti

NMG = numero di malati gravi guariti

NMN = numero di malati normali guariti

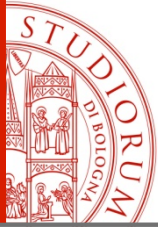
NMA = numero di malati che hanno atteso in coda

NTOT = numero totale di malati simulati

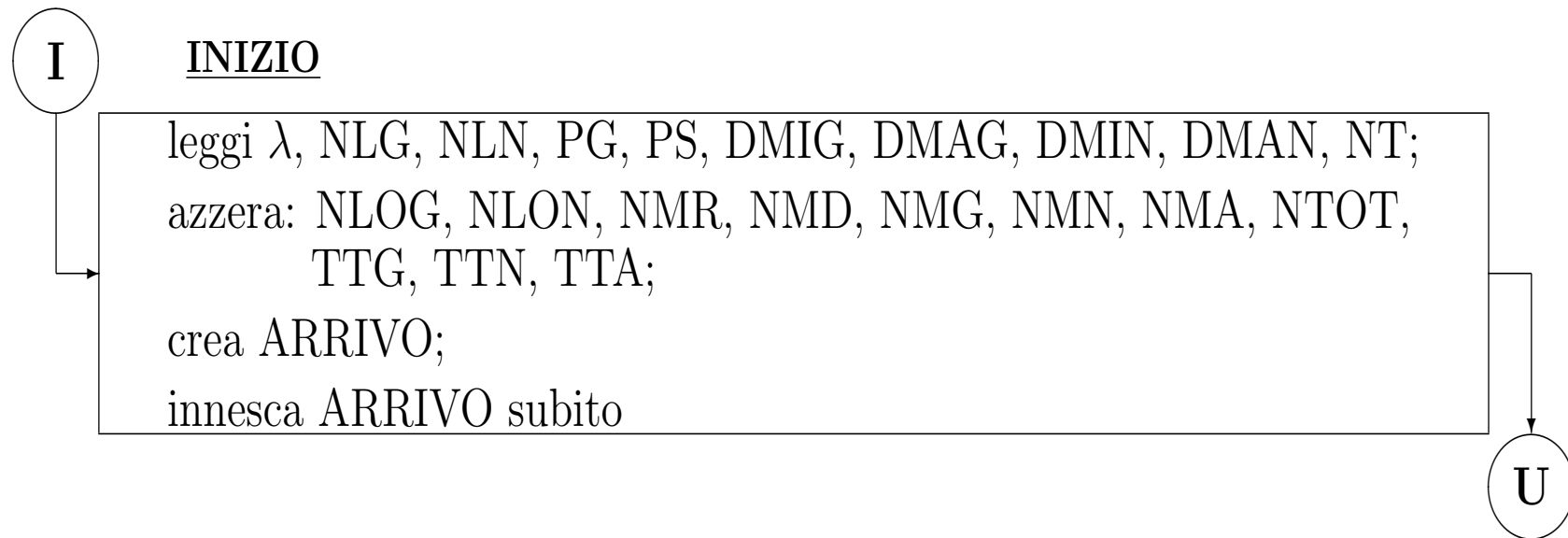
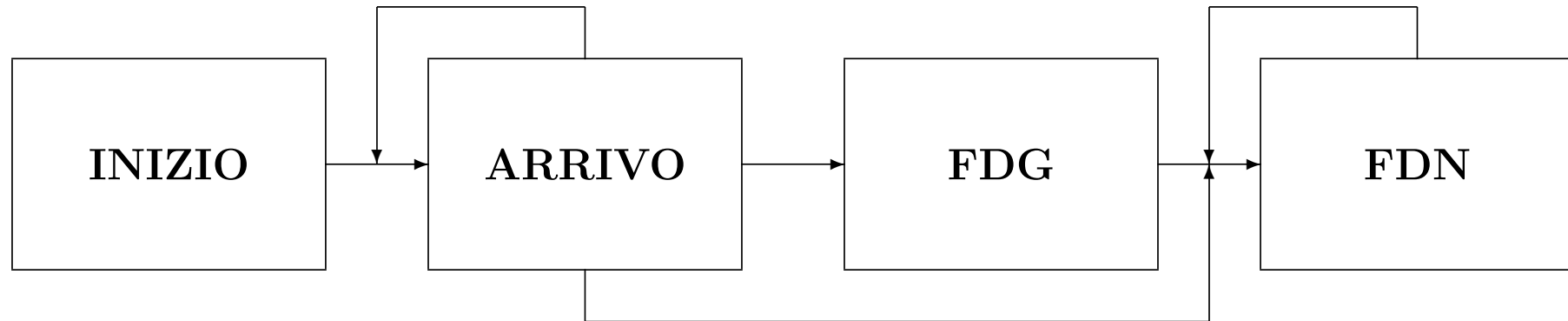
TTG = tempo totale trascorso dai malati gravi nel sistema

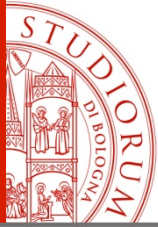
TTN = tempo totale trascorso dai malati normali nel sistema

TTA = tempo totale trascorso dai malati in coda

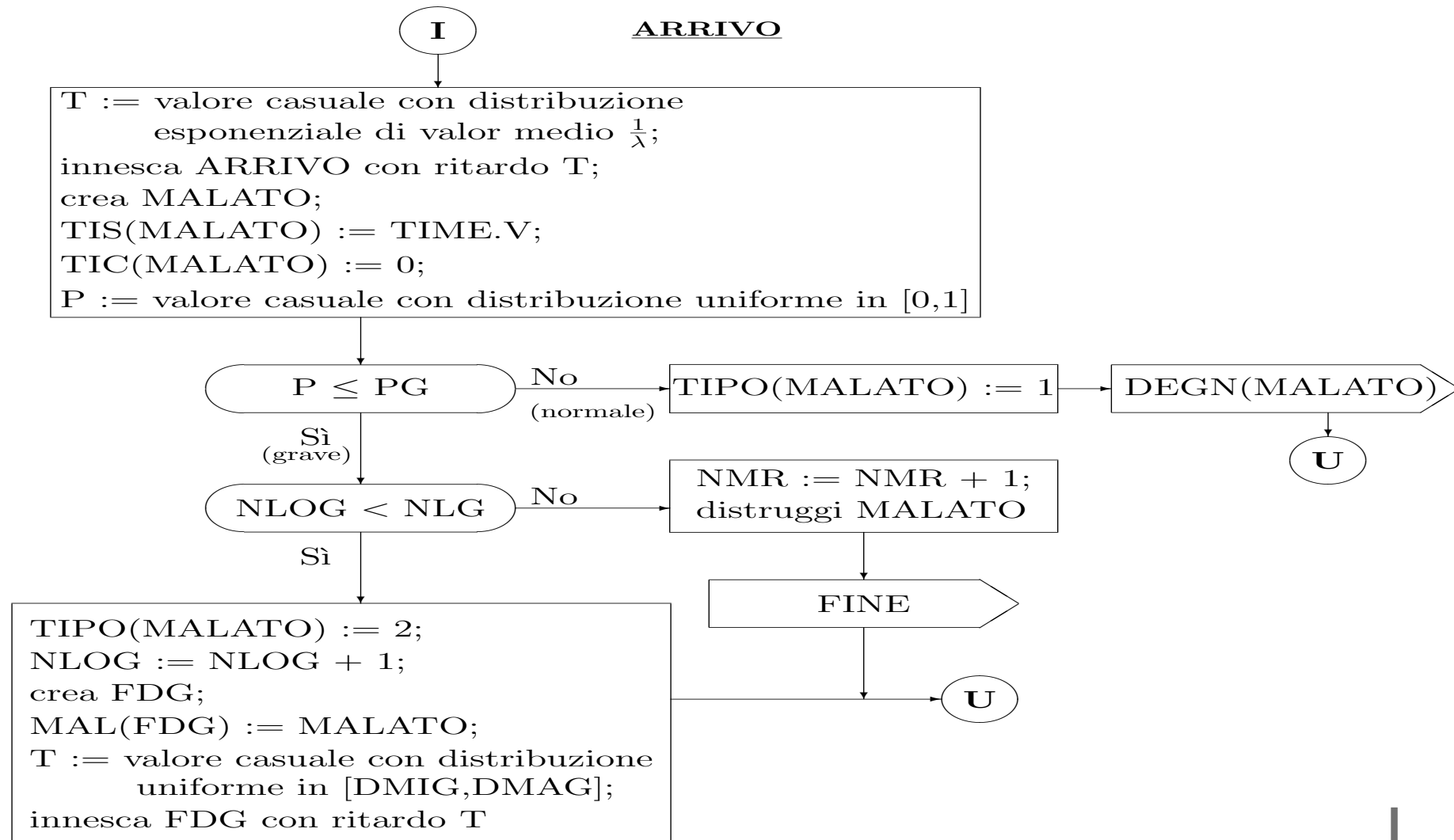


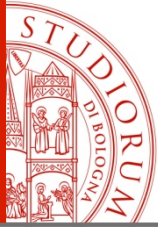
Ospedale, diagramma di flusso (1)



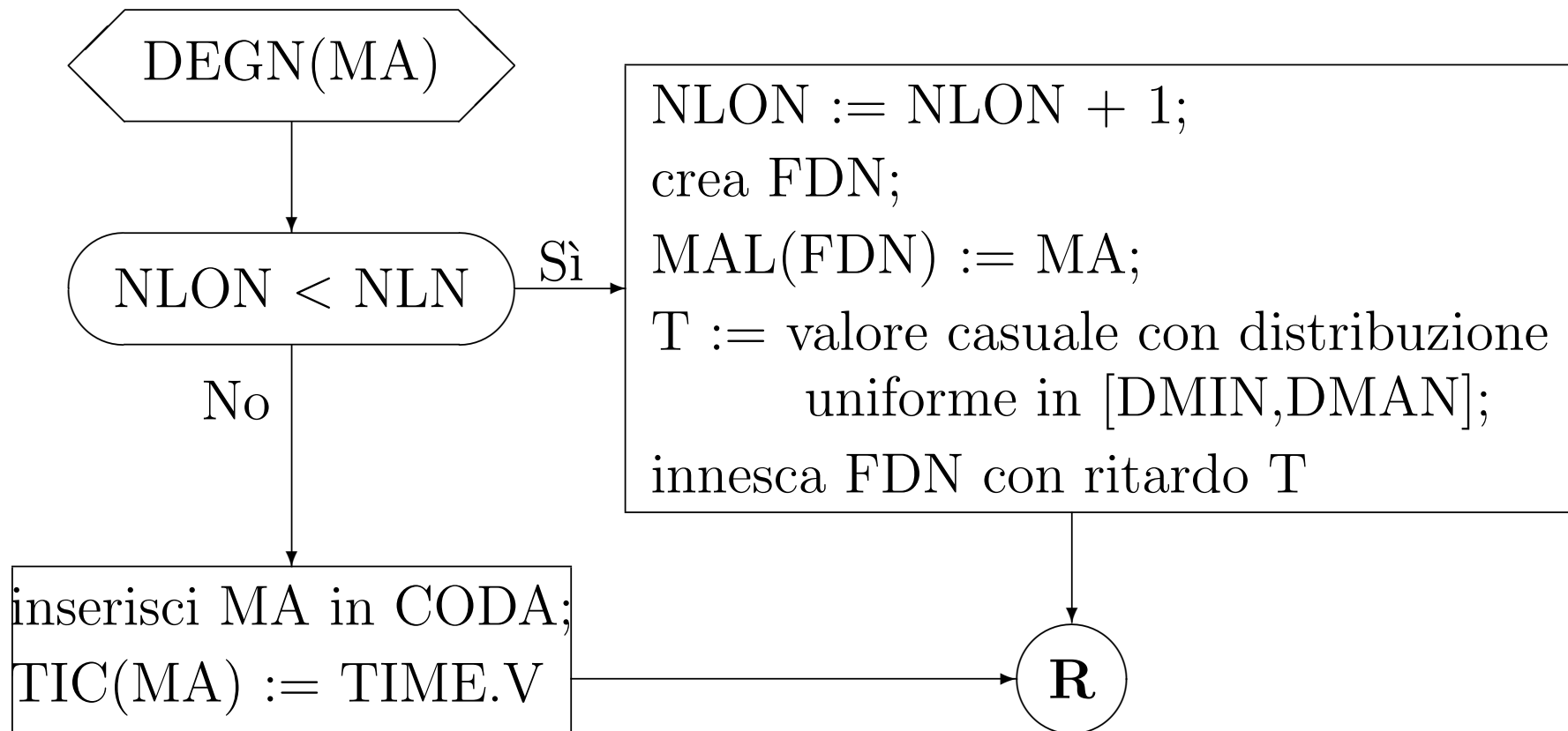


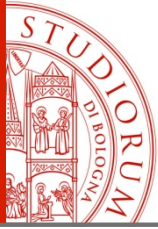
Ospedale, DdF (2), Arrivo



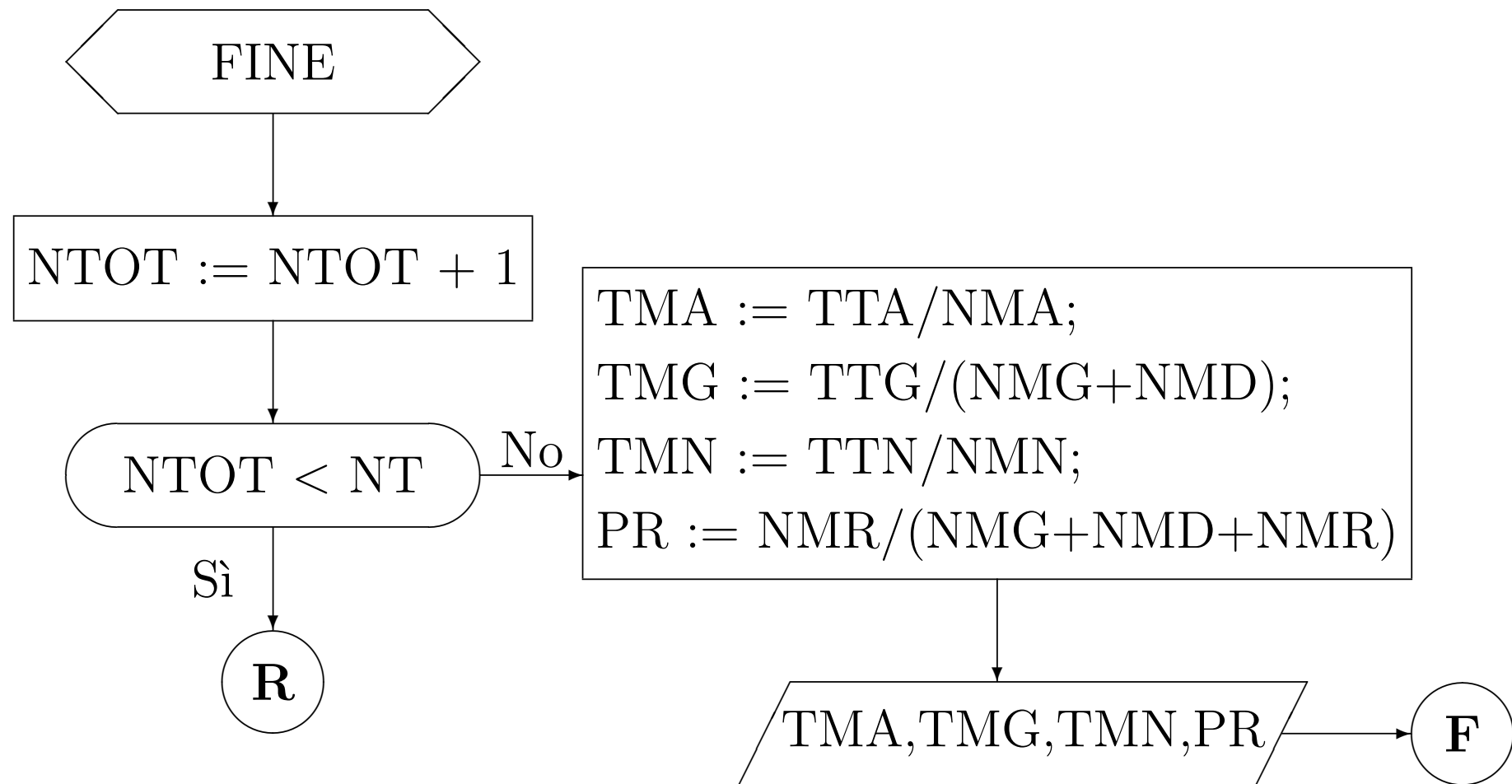


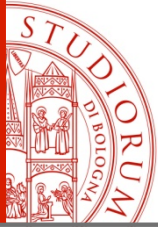
Ospedale, DdF (3), proc. DEGN



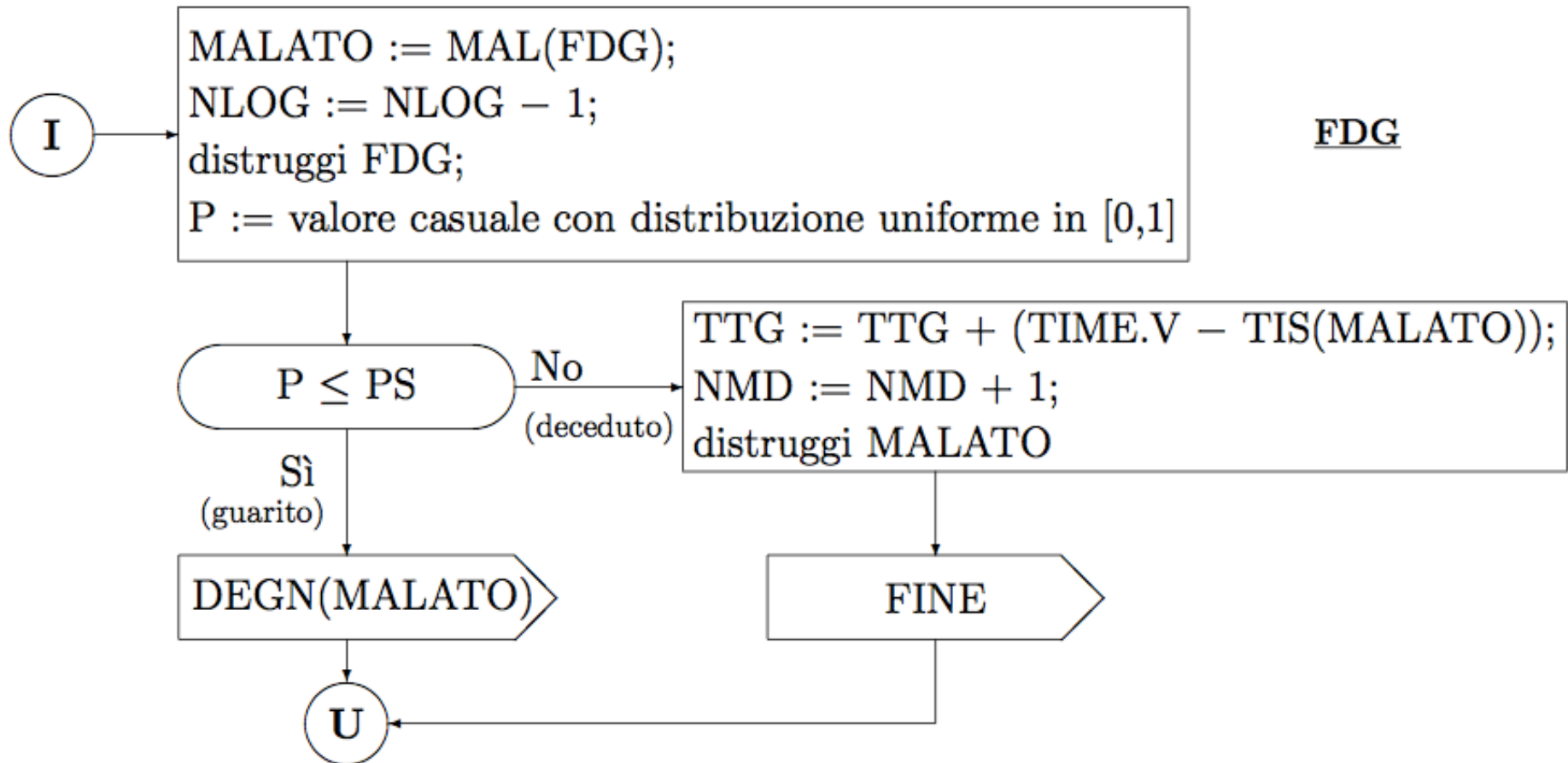


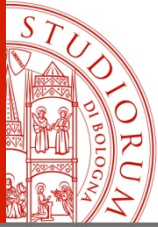
Ospedale, DdF (4), proc. FINE



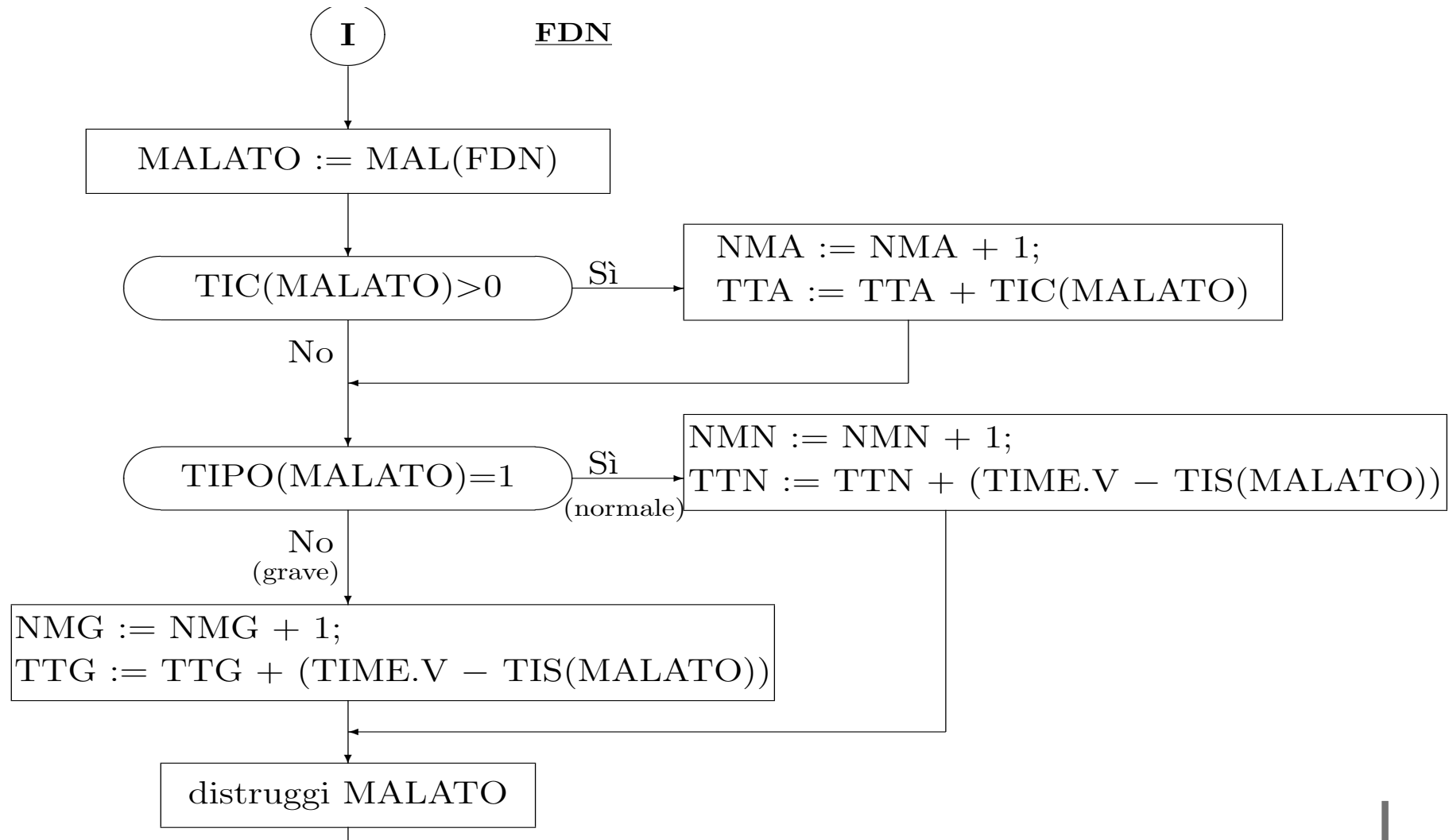


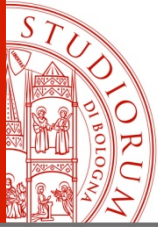
Ospedale, DdF (5), FDG





Ospedale, DdF (5), FDN





Ospedale, DdF (5), FDN (parte 2)

